Mobile cloud middleware: Towards selective cloud services based on location of users

Hind Bangui, Said Rakrak, Said Raghay

Applied Mathematics and Computer Science Laboratory
Cadi Ayyad University
Marrakesh, Morocco

Abstract
Due to the advancement of cloud computing, mobile technologies are growing at a very high pace since cloud platforms provide their own resources to meet the needs of mobile users. However, the combination of cloud computing and mobile computing is created not only to consume remote services, extend the capabilities of mobile devices or profit from all the massive advantages of the cloud computing, but also to enable mobile users to exploit context information provided by their smartphones. Hence, context-aware systems offer entirely new opportunities for mobile cloud applications by means of collecting mobile users' context data and adopting cloud services' behaviors accordingly. In the previous paper [1], we introduced the detailed architecture for a middleware system, which can define a new interaction between a mobile client and different cloud providers. The current paper is an extension of the earlier one describing an improved version of the proposed middleware, which exploits available cloud resources as well as mobile users' locations to better finding effective mechanisms to consume the right and light cloud services. We conduct experiments using cloudAnalyst to prove the feasibility of our proposed architecture.
Keywords:
Mobile cloud computing; middleware; cloud computing; mobile computing; context-awareness.

1. Introduction

Nowadays, smartphones have a significant impact on humanity's daily life ever since they have incorporated a variety of categories such as healthcare and social networks' applications. In fact, these small gadgets present several advantages, amongst which we emphasize the ease of altering from one network to another and the robustness of applications as they can get quick results to mobile clients. Due to these benefits, mobile devices have become the primary computing interface in the world of IT and commerce.

Currently, more and more applications have already been migrating from traditional personal computers to smart phones. One of the impacts of mobile devices is BYOD concept "Bring Your Own Device", which aims to attract industries to investigate in mobile computing in order to reduce the desktop execution power. Consequently, the employers can use their personal devices as elements of work, and become more productive by means of easy deployment procedures, which can be performed with low cost regardless of the time and location. However, the first screen in the world being able to capture contextual data is still resource constrained with limited computation power, memory, storage, and energy. Accordingly, these challenges have slowed down the growth of mobile computing that has become a force to be reckoned today.

Enhancing the growth of the service quality of mobile devices requires considering external resources. One of such is cloud computing, which provides various services for hosting the mobile applications' delivery and extending their capabilities in storage, computing as well as energy. Besides, due to the simplicity, flexibility, availability, and advancement of cloud computing as being one of the most important technology in this decade, different kinds of clients (i.e. mobile devices) are aiming to make use of the cloud so as to meet their every day's needs. Therefore, the recent emergence of cloud computing and mobile computing is described as multiple services of cloud computing that are available in mobile environment.

The study in [2] shows that mobile cloud traffic will grow 11-fold from 2014 to 2019, a CAGR (Compound Annual Growth Rate) of 60 percent (figure 1). As a result, the development of mobile cloud applications has recently received significant attention in academic researches and motivated industries to conduct in-depth mobile cloud computing investigations. However, mobile cloud clients cannot be compared to the traditional cloud clients as the consumption of cloud services is affected by the users' mobility and bandwidth stability. Accordingly, mobile cloud computing is extending cloud computing to enable mobile users not only to access rich and remote services but also to enable them to benefit from context information offered by their smartphones. This latter has received more attention in recent years since mobile devices and computers are equipped with more accurate sensing capabilities than before. As a result, the impact of integrating mobile devices' context in mobile cloud computing is needed to create new categories of cloud services. For example, m-healthcare is the most emerging

Manuscript received September 5, 2017
Manuscript revised September 20, 2017

mobile applications, where mobile devices can sense the activities of mobile clients and the surroundings. The sensed data is collected and analyzed by remote cloud servers to estimate the health conditions and evaluate the environment of mobile clients. Put differently, providing context information allows more providers to incorporate context-aware capabilities in their cloud services. Moreover, given that many services are available with the same functionality in various locations, mobile cloud applications can actually consume nearby or distant cloud services from multiple cloud sources such as mobile cloudlet [3] and datacenters. Finally, after acquiring clients' context information, the ultimate goal for each cloud source is to offer the best service regarding user's requirements.

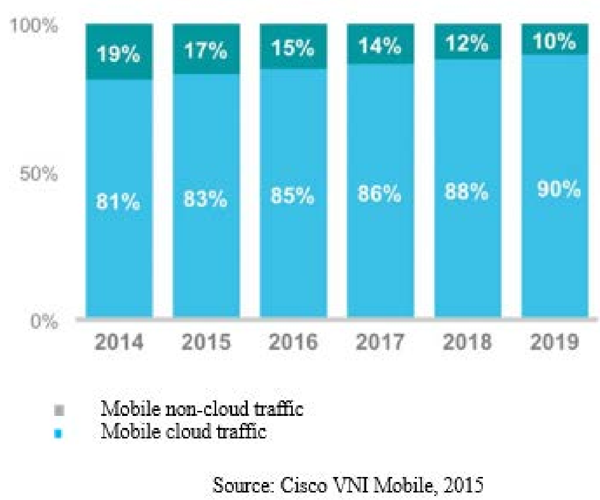


Fig.1 Cloud Applications Will Account for 90 Percent of Mobile Data Traffic by 2019[2]

Actually, satisfying the mobile clients is challenging due to the mobility nature of devices. Mobile devices move through different geographical locations with diverse or seamless connectivity network. Yet, the cloud services can be affected by long latency. In addition, each cloud resource is mainly responsible for users nearby. However, the users and cloud resources (i.e. mobile cloudlets) may change their locations and become disconnected from each other, which may result in cloud service unavailability. Another case presented by the study in [4] that describes how the triple contexts (country, industry and firm) affect customers' views of corporate brands. Besides, it compares between global and local services (Google as a global brand and Baidu as a dominant Chinese brand). Another example of local services is Yandex offering several services and performs very well in Russia than Google. Still, its performance is very poor outside Russia because it is built to understand local culture and Russian netizens [5].Yet, a cloud resource can satisfy mobile users in a certain moment. Lastly, in order to create ubiquitous mobile cloud service, it is necessary to consider the location of mobile users.

The above mentioned problems generate the following challenges regarding users' geographic information:

- How to integrate location based information in mobile cloud computing?
- From available and similar functionalities based cloud services, how can mobile users select their proper services according to their preference cost and geographic information?
- Considering users' behaviors, how to evaluate and exploit users' services access history to select the best cloud resource that gives the most accurate results?

In the previous paper [1], we introduced the detailed architecture for a middleware system, which can control the communication between mobile clients and different cloud providers. The current paper is an extension of the first one; it describes an improved version of the presented middleware, which focuses on the user's presence, his location and environmental information. The goal of this paper is to allow mobile clients to overcome their limitations by providing effective ways in order to select a cloud service. For example, requests' dynamic adaptation according to available cloud services and devices' context information, especially their locations, offers more accurate information to better choose the suitable service. Taking the advantages of existing cloud resources and users' location, the middleware will reduce execution time of mobile applications and battery consumption by selecting an optimal cloud service from the most adequate and the nearest cloud resource. Further, it aims to optimize results before delivery to mobile clients.

The remainder of the paper is structured as follows. In section II, we present some important related works. the motivation of research is presented in the section III. Next, section IV describes the improved version of our middleware architecture with the description of the interaction between its modules. In section V, we conduct experiments to evaluate the performance of our proposed middleware. Section VI concludes the paper and some possible future works.

2. RELATED WORK

Mobile cloud computing is born to bring new kinds of services to mobile users and utilize cloud computing to deliver ubiquitous mobile service access. In [6], the authors clearly present the applicability of mobile cloud computing to a wide range of mobile services.

Additionally, they describe how mobile cloud is increasing in different fields of our daily life; for example, m-healthcare which facilitates efficient patient treatment for medical consultation.

Some of the typical mobile cloud applications are discussed in [7] to prove the lack of mobile specific infrastructure as compared to the existing cloud technologies. Basically, the authors present the cloud computing technology as a new external source for mobile device to overcome the smartphones constraints. However, the traditional smartphone application models do not support the development of applications that can incorporate cloud computing features and demands specialized mobile cloud application models. Hence, the survey focuses on identifying the parameters that affect mobile cloud application models. In addition, it compares the application models along with their critical outstanding issues. Finally, it suggests future research challenges and opportunities. So, mobile cloud applications represent one of the newest developments in cloud computing advancement. However, these kinds of applications necessitate employing a middleware, which can enhance the integration of the cloud computing into the mobile environment. Thus, the use of middleware has attracted many researchers in the recent few years.

As previously mentioned, mobile cloud technologies are evolving to overcome the computing, storage, and power limitations of mobile devices. The framework, which is proposed in [8], takes advantage of cloud computing to overcome the limited resources of mobile environments. Moreover, it aims to augment the capabilities of mobile devices in order to become reliable service providers. The middleware in [9] also aims to boost the mobile device real-time communication capabilities to mobile nodes.

As for [10], the authors propose a middleware that allows developers to easily construct applications for mobile cloud environments. This integrated middleware platform supporting transparent distributed deployment and scaling among multi-users' applications and cloud infrastructures.

Another approach to overcome the limitation of mobile devices' storage is cited in [11], where a light- weighted storage middleware, namely user-oriented Mobile File System (uMFS), is implemented to improve the performance of massive-data storage service for mobile end-users.

The main objective of [12] is to describe how smart mobile devices have changed users' preferences for computing, and replaced a number of portable computing and communication devices as all-in-one device. In fact, the authors explain the different techniques to augment smart mobile devices resources based on resources available within the cloud. Likewise, they discuss the issues in current DAPFs (Distributed Application Processing Frameworks) and challenges to optimal distributed application frameworks for mobile cloud computing. One of the important perspectives of cloud based application processing is homogenous and optimal distributed platforms. This challenge aims to provide homogenous solution for heterogeneous devices, operating platforms and network technologies with minimum possible resources utilization on the smart mobile devices.

A very interesting research has been conducted in [13], where the authors present a novel standard bridge between the mobile terminals and the cloud. Unlike to traditional cloud computing, that has been made to provide adaptively to different mobile devices, the architecture introduces a novel cloud terminal adaptation platform based on mobile web-middleware, which acts as a standard interface for the mobile middleware and the cloud.

Similarly, MiLAMob [14] is another middleware-layer for authenticating which allows mobile consumers to access Amazon S3. This later is a simple cloud storage services that uses Web Services interfaces based on REST and SOAP to store data. This middleware aims to simplify the authentication process in real time and with minimal HTTP traffic. Furthermore, MiLAMob employs the OAuth 2.0 technique through the social networking sites (i.e. Twitter, Facebook, Google+) to identify the end-user and use security tokens to handle the tedious authentication process that mobile browser-based apps encounter when consuming Amazon S3 data.

In typical augmentation approach namely resource-oriented mobile cloud (RMCC) [15], the adjacent service-based mobile cloudlets host prefabricated Restful services to be asynchronously called by mobile service consumers at runtime. The Restful mobile cloud computing framework is the first Restful service-oriented solution that offers platform-independent RiMA (resource-intensive mobile applications) execution in mobile cloud computing with time-energy saving.

In [16], the authors present MCM (Mobile Cloud Middleware) as an intermediary between the mobile device and the cloud in order to address the issues of interoperability across multiple clouds as well as transparent and asynchronous delegations of mobile tasks to cloud resources. Moreover, MCM can previously develop a customer service based on its composition, which reduces the number of offloading time needed in a mobile cloud application. However, MCC cannot provide a dynamic service provisioning according to mobile application needs.

A very interesting research has been presented in [17], where the authors present a monitoring middleware for cloud services. The suggested design aims to monitor the

SLA compliance and provide a report containing SLA violations. As a result, the cloud providers can make decisions regarding how to improve the quality of cloud services in order to satisfy their customers. However, the authors did not apply the monitoring middleware on different cloud platforms; consequently, they did not analyze the interplay of service quality parameters and SLA requirements across multiple models like IaaS.

Another middleware [18] uses skyline query system to discover preferable services in the mobile cloud system. By exploiting user's service access history, the proposed distributed skyline query processing can reduce networking communication cost in the mobile cloud system while preserving the mobile client satisfaction.

Similar to other researches, the middleware in [19] can store the copies of service results, and return directly its cached data to the mobile clients. Hence, the proposed design can reduce response time and increase the lifetime of mobile devices.

In fact, the previous middleware architectures benefit from all the advantages of the cloud computing; whereas, they do not take advantage of mobile devices by deploying their current context. For that reason, the authors, in [20], provide a complete survey on the field of mobile cloud computing, especially context-awareness, in its state of art applications, research challenges, opportunities and future research directions. As a matter of fact, context-awareness becomes the subject of recent fields of research in pervasive computing, where the applications are able to change and adapt the behavior according to their users' situation. For example, 'Mobility' is an important user context piece of information as the mobile users should be able to perform their tasks uninterrupted despite of their position. Additionally, mobile clients can use provided tools by mobile computing willingly and when it is needed regardless of their movements, which facilitate location flexibility. Thus, it is important to benefit from such context information for mobile cloud applications' improvement.

3. motivation of research

A. Context-aware:

Unlike previous solutions, we propose in this paper a middleware defining new interaction model between mobile devices and cloud services; besides, we take into account the mobile user's context information. Put differently, the middleware tries to use mobile device context so as to adapt its behavior and present customer cloud services to mobile users. Finally, the introduced middleware interaction methods depend on the users' context such as location, bandwidth and remaining energy.

As context awareness has become the subject of recent fields of research in pervasive computing, where the applications change their behaviors according to users and their situations, it is wiser to benefit from such information to enrich mobile cloud services. In fact, the research about this concept in computer science can be divided into two main groups: The first one contains researches interested in incorporating the context to their applications to enhance the services and performance offered by their approaches. Whereas, The second category, to which our approach belongs, regroups researches interested in context as a concept looking for ways to cover it computationally through models, frameworks and methodologies.

Truthfully, it is challenging to define the word 'context' and many researches tried to find their own definition for what context actually includes. Hence, the context is defined in many perspectives as follows:

Firstly, the authors, in [21], propose a definition of the context as: "Context is the set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user".

Secondly, the author elaborate in his most cited definition for context: "Context is any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that is considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity and state of people, groups and computational and physical objects."[22]

Last but not least, the authors, in [23], introduce another definition in which they define context as: "Context is a set of the external parameters that can influence the behavior of an application by defining new views on its data and its available services. These parameters may be dynamic and may change during the execution."

On the other hand, the appearance and integration of mobile devices has led to the introduction of pervasive systems which increasingly have become popular. First launched by Weiser in [24], the term 'pervasive' refers to the complete integration of devices into the user's every-day life. In 1991, Weiser said that "the most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." This was a vision too far ahead of its time because the hardware technology needed to achieve did not exist yet. In the last years, this vision has become a reality.

Accordingly, the notion of context should be well thought-out in order to concretize this type of systems. The context is any information that can be used to characterize the situation of a person, a place, or an object that are considered relevant to the interaction between a user and an application. These latters may include the quality of service of a network, user's preferences, location, time, and language, etc.

As a result, there are several definitions for context, but, it is clear that context information helps the applications to make the right decision at the right time. For example, if a mobile client wants to play a video in his mobile device, a context-aware video playing application on the mobile device may automatically select the appropriate variants according to its available resources (i.e. memory and battery power), then sends this information to the cloud provider. So, the provider will be able to give a suitable video stream which respects to the capabilities and resources available for the mobile client. In this case, context awareness aids cloud providers to adapt dynamically their services according to the continuously changing context of the mobile user.

B. Location of mobile user:

With the development of smartphones in recent years, diverse mobile applications (i.e. Foursquare [25] [26]) can identify whether or not a position corresponds to a mobile user. The location, which characterizes each mobile client, has become the most important term in mobile cloud computing. In addition, it is exploited as a context-aware service used to deliver personalized mobile services according to the elements of the environment surrounding the users. In the other words, a mobile client can use one of such multiple available cloud resources to select adequate cloud service. On the subject of traditional consumption, central cloud resources are always ready to offer unlimited elastic resources remotely. Besides, there are many interesting works involving mobile devices as small computing unit in cloud environment as illustrated in [28]. Whereas, a cloudlet in [29], which acts as a micro-cloud deployed in public places, is another source of customized cloud assets aiming to reduce communication delay and satisfy the closer mobile client. In addition, the mobility concept, which characterizes the mobile devices, is exploited to produce another form of cloudlets nodes entitled as mobile cloudlets (such as introduced in [3]). These mobile cloudlets exploit mobile devices (smart phones, tablets and others) to speed up the accessibility to customized cloud services and increase the execution time.

A very important study in [27] describes how can we exploit the movement of people alongside their activities to dramatically improve human movement prediction? This study demonstrates that exploiting accurate context of mobile applications is far better than exploiting the past history of mobile clients. Yet, mobile applications are able to determine the mobility forecasting of mobile clients and, accordingly, provide personalized services relating to their context. For example, Baidu's Search performs very well in China then Google's Search. However, its performance is very poor when a user uses this service outside China because it is built to understand local culture and Chinese netizens. Similarly, Yandex covers many aspects of Russian daily life. Another internal service is Naver that dominates as local service in South Korean. That means, there are many services that are preferred and used as the main service within a specific region. Yet, a preferable or local service could satisfy a mobile user in one moment, but fail in another moment due to the change of users' context. The same applies to the copies of cloud service results which are stored in the middleware as described in [19]. Hence, it is important to know mobile customer's context, especially location, in order to bring him/her a better service. As a result, we propose a middleware that can adapt to dynamic nature of mobile devices, and prove how context information can be used in a beneficial way. The proposed middleware will take into account users' contexts which form the key for creating a new behavior of the mobile application services. In addition, the middleware is based on three elements: Adaptation, caching, and optimization, not to forget considering context information to assess the selected process of the cloud service. Since the service discovery in mobile cloud computing is very essential to provide the mobile services, the middleware must incorporate adaptability and collect context information from mobile devices in order to allocate the best cloud service. Meanwhile, we take into account the advancement of mobile cloud platforms, a nearby service provider can be found correctly by providing mobile users' geographical positions. Meaning, the proposed middleware gathers available distinct or similar cloud resources which can satisfy users' requests according to their requirements. Then, it visualizes the information offered by the cloud providers especially service location, and it exploits users' positions to decide which is the best cloud service to deliver the corresponding responses (figure 2).In case a user's service access history is saved, the middleware makes the evaluation of each service whether or not it is suitable considering the user's context, which may reduce response times while satisfying the mobile user's real time demands.

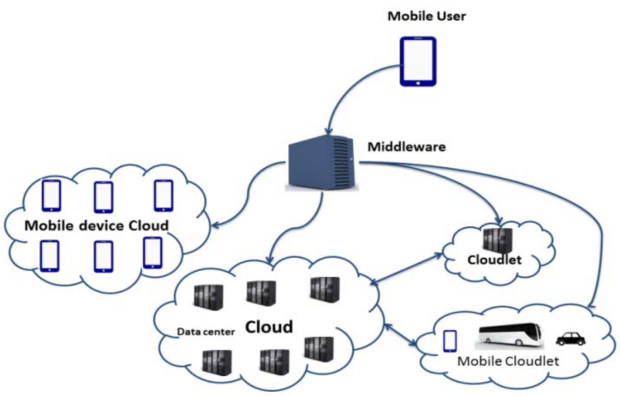


Fig.2 Middleware for multiple cloud resources

4. Mobile cloud middleware architecture

Here we present a detailed architecture of a middleware system, which can handle actual interaction with cloud web services in different locations and meet the user's requirements. Our solution is composed of many components (figure 3), which control the communication and aim to perform interaction between the mobile client and different cloud providers.

When a mobile application delegates a mobile task to the cloud, it sends a request to the handler and the context monitoring module intercepts the current context of the device client. The request is processed by the manager; it first creates a session assigning a unique identifier which is used for handling different requests as well as for sending the notification back when the process running in the cloud is finished. Next, the manager sends the request to the caching module to check if it is saved in the system.

First case, in which the request is not found, it is forwarded to the adapter request module. This latter handles the service requests according to the current context of the mobile device provided by the context monitoring module and the QoS level.

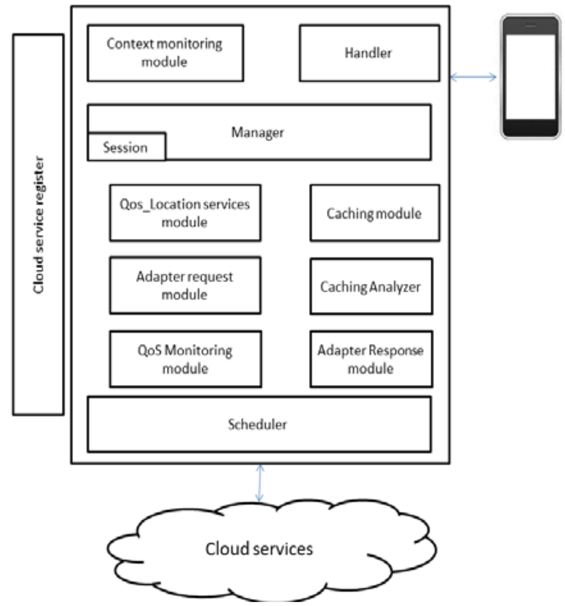


Fig.3 The middleware modules

The QoS level is selected from the QoS-location services module which contains service lists on the cloud and their QoSs in different locations, this module will be updated by considering mobile users location. Besides, the adapter request module exploits the estimated completion time for each resource to select the adequate QoS level; we present the algorithm (1) in this section that is applied by the middleware.

The Max-Min Algorithm
1. for all submitted tasks in meta-task; T_i
2. for all resources; R_j
3. $C_{ij} = E_{ij} + r_j$
4. While meta-task is not empty
5. Select task T_k with the maximum completion time.
6. Assign T_k to the resource R_j which gives minimum completion time.
7. remove T_k from meta-tasks set
8. update r_j for selected R_j
9. update C_{ij} for all i

The algorithm (1) formulated in this paper makes use of the concept of Max-Min scheduling [30] approach for the purpose of calculating the processing time or expected completion time to the available cloud resources. In Max-Min algorithm, the expected completion time (Cij) is the sum of the estimated execution time (Eij) of task (Ti) on resource (Rj) and the time to become ready to execute a task (Ti) which is denoted by rj. In our proposed algorithm, the expected completion time consists of estimated execution time, transmission time (Tt), and ready time together (1).

$$\text{Estimated completion time} = E_j + r_j + T_t \quad (1)$$

$$\text{Where } E_j = \frac{\text{delegated task}}{\text{computing capabilities}} \text{ and}$$

$$T_t = \frac{\text{Volume of data to be transferred}}{\text{Bandwidth from the client to the cloud resource}}$$

Algorithm 1

1. Start discovery service
2. Gather nearest cloud resources according to mobile users' position
3. For all cloud resources R_j, estimate the completion time of the task.
4. $C_j = E_j + r_j + T_t$
5. // r_j is the time to become ready to execute a task
6. // E_j is the estimated execution time of task
7. // T_t is the transmission time
8. If user specifies cost $_{preference}$ then
9. Estimate expected cost for the estimated completion time
10. If cost $_{expected}$ - Cost $_{preference}$ >0 then
11. Remove R_k from list of cloud resources
12. End if
13. ;End For
14. Classify the estimated completion time from high to low level
15. If user specifies cost $_{preference}$ then
16. Assign request to the resource R $_{candidate}$ which gives cost $_{preference}$=> expected cost
17. Else if assign request to the resource R $_{candidate}$ which gives minimum expected completion time

First, the adapter request module determines the estimated completion times (ECT). Next, it classifies ECT from high to low level. Taking into consideration the previous estimated results, the adapter provides the estimated costs for each ECT. In case the mobile user specifies his preference cost, the adapter compares it with the expected costs. Then it selects cloud resource which gives the equal or approximate expected cost according to the user's preference cost. Second case, if the client does not define any cost, the adapter decides to select the cloud resource with the minimum expected completion time.

After the selection of adequate cloud resource, the QoS monitoring module monitors the QoS level. Any deviation is asserted at the QoS levels of service offered by the service provider or the context forwarded by the mobile client; then, the context monitoring module or the QoS monitoring module advertises the adapter request module to re-evaluate and rebind to a more appropriate service according to the context and QoS level. Then the adapter request is sent to the scheduler which, by itself, sends the request to the provider in the cloud.

Once the middleware receives the response, it is sent to the adapter-response module. The adapter-response transforms SOAP messages to Restful and XML formats to JSON, and optimizes the results by extracting relevant information. Finally, the optimized response is forwarded to the manager. Simultaneously, the caching module saves the original resource of the selected service, real response time, cost, and estimated completion time on which the middleware was estimated before sending the request to the cloud. Finally, the manager sends back the result to the handler and notifies the mobile client by using the asynchronous notification.

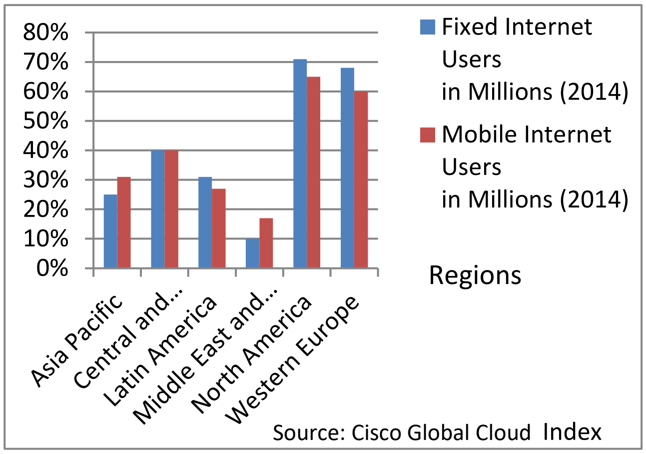
Second case, if the request is saved within our middleware (algorithm 2), the caching module retrieves all the services with their cost, real response time and estimated completion time on which the middleware was estimated before sending the request to the selected cloud resources. Then the caching module sends the results to the caching-analyzer. Regardless of the performance of cloud service, some users demand high quality of cloud services to execute their request, whereas others can be satisfied with low quality of cloud services according to their preferable cost. So, considering cost and response time should lead to a better cloud service. First, the caching-analyzer compares the estimated completion time with the real response time of different services. Then it initializes two lists that contain the high QoS and low QoS, respectively. Next, it selects the best service, which gives the earlier finished execution, according to the user's context. If suitable service not found in the tow lists, the caching analyzer module sends the request to the adapter request module in order to satisfy the user.

Algorithm 2

1. For all services save in caching module
2. Select services corresponding to the request and mobile user's position; list-S_i.
3. For each service in list-S_i, compare real response time T_r and estimated completion time T_c
4. Classify services into two lists; high-list and low-list
5. While high-list is not empty
6. for each cloud service with high QoS (T_r<=T_c)
7. If user specifies cost $_{preference}$ then
8. Select service's cost
9. If cost - cost $_{preference}$ >0 then
10. Remove S_k from high-list
11. End for
12. If user specifies cost $_{preference}$ then
13. Assign request to the service S $_{candidate}$ which gives cost $_{preference}$=> cost
14. Else if assign request to the service S $_{candidate}$ which gives earlier finished execution
15. Repeat
16. While low-list is not empty
17. for each cloud service with low QoS (T_r >T_c)
18. If user specifies cost $_{preference}$ then
19. Select service's cost
20. If cost - cost $_{preference}$ >0 then
21. Remove S_k from low-list
22. End for
23. If user specifies cost $_{preference}$ then
24. Assign request to the service S $_{candidate}$ which gives cost $_{preference}$=> cost
25. Else if assign request to the service S $_{candidate}$ which gives earlier finished execution
26. Repeat

5. Experimental results

Given a delegation service request in mobile cloud, it is a challenging issue to determine whether or not and where to delegate the tasks among mobile devices. Actually, several cloud providers are available with the same functionality, each one offering different pricing models (pay-as-you-go) and located in different geographical locations. Thus, mobile applications can benefit greatly from cloud infrastructure services, which have become cheaper and easier as they use flexible and elastic infrastructure services. Further, mobile users can minimize costs and obtain nearby or distant services. In contrast, there are several factors that impact the quality of service to be wisely observed. Some of these factors include geographical distribution of users' bases, and capabilities of the Internet infrastructure within those geographical areas, amongst other factors. Cisco offers the regional Internet access penetration for 2014 and 2019 (figure 4).



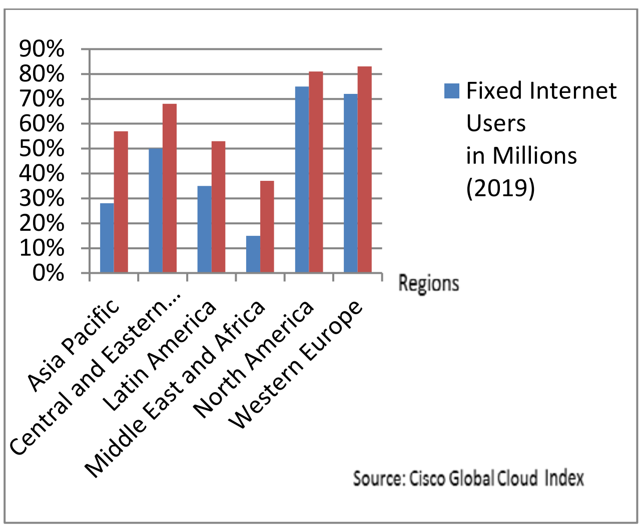


Fig.4 Regional Internet Penetration (Percentages Indicate Users with Internet Access per Region) in 2014 &2019

On the other hand, mobile devices use various sensors to catch the real-time situation of the users. So, if we exploit the mobile environment especially location of mobile users, we can minimize the impact of those factors and take them into account in the selection of cloud service. Put differently, if we minimize the response time, we will improve the quality of services supplied by cloud computing and, at the same time, we will increase the lifetime of mobile devices; thereby, energy consumption will be reduced.

C. Simulation Tool:

The process of evaluating the middleware requires the use of CloudAnalyst [31] that allows cloud applications to determine the best strategy. This latter is based on selecting cloud services in terms of adaptation, dynamic reconfiguration, or other criteria. Also, the underlying architecture of CloudAnalyst is built on top of CloudSim [32] with an additional graphical representation. Actually, CloudAnalyst is very easy to use and has the ability to produce the output in graphical form. Additionally, users can easily repeat experiments with the same and/or different parameters (table 1) and view graphical results.

Table 1: Characteristics Of CloudAnalyst

Characteristics	Description
Region	6 regions correspond to 6 main continents in the world.
User base (UB)	UB represents a group of users considered as a single unit in the simulation.
Internet	It determines the characteristics of the Internet that is used to simulate the data transfer, including the latency and bandwidth parameters.
Cloudlet	It represents a single or set of user requests.
Data center (DC)	It defines in detail the characteristics of computing servers, which are either heterogeneous or homogeneous in nature based on their hardware configurations (i.e. allocating bandwidth, storage, memory and speed of CPU)
Service broker	It decides as to which DC should be selected to provide the services to the requests from each UB.

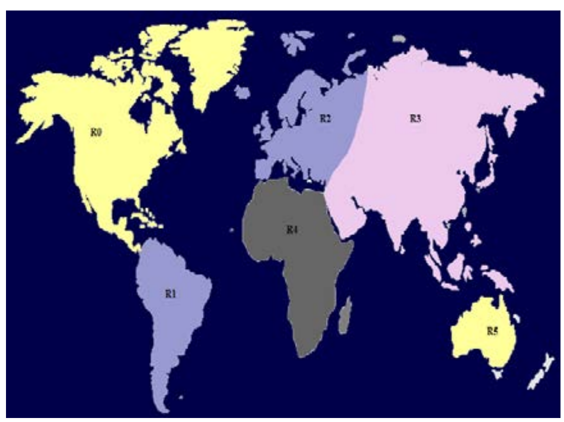


Fig. 5 Regions in CloudAnalyst

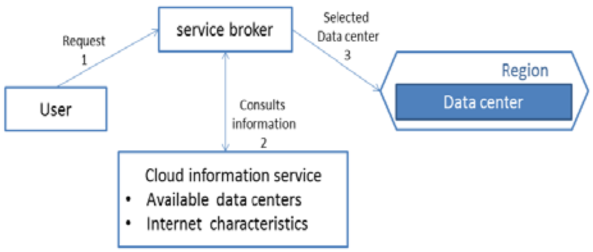


Fig. 6 Service Broker

One of the main components of CloudAnalyst is Service Broker that controls the traffic routing between Users' Bases and cloud resources (figure 6), which are located in different regions (figure 5); CloudAnalyst defines three types of service brokers that decide which cloud resource should treat the service to the requests. The three types of service brokers are listed as follows:

- **Service Proximity Based Routing:** First, the service broker determines the region of users' bases. Next, the region proximity list is ordered depending on the network latency (lowest to highest).Then the cloud service is selected and used for processing.
- **Performance Optimized Routing:** the traffic to the cloud service is selected according to the performance of all cloud resources.
- **Dynamic Service Broker:** Increases and decreases dynamically the number of virtual machines allocated in the cloud resources.

In our case, we aim to implement Algorithm 1 & 2 in Service Broker in order to select a suitable cloud resource based on users' locations (or region as named in CloudAnalyst) in order to attain significant performance improvement. For that reason, we set the parameters of CloudAnalyst in two scenarios according to the proposed algorithms.

D. Scenario 1:

As a first experiment, we compared with the traditional Max-Min algorithm, and we focused to select data center according to the performance of all cloud resources (the faster the best).The second step is to integrate Service-location-Service Broker (Algorithm 1) which determines the region of users' bases and then asks for the region proximity list regarding the requesters' regions. The list is ordered according to the client's location (nearest to farthest). In all our experiments, we have set the parameters for users' bases configurations and data centers configurations used in simulation as shown in table 2 and table 3 respectively.

The locations of users' bases have been defined in five different regions; the transmission delay between regions is defined in figure 7. We have taken three cloud resources to handle these users' requests. One cloud resource (DC1) is located in R0 (region 0), DC2 in R2, and DC3 in R4. Moreover, it is very important to state that DC1, DC2, and DC3 have different performance capabilities.

Table 2: User bases

Name	Region	Data Size per Request(bytes)	Request per User per Hr
UB1	0	100	60
UB2	1	100	60
UB3	2	100	60
UB4	3	100	60
UB5	4	100	60

Table 3: data center configuration

Parameters	**Values**
Regions	DC1: R0 DC2: R2 DC3: R4
Architecture	X86
OS	Linux
VMM (Virtual Machine Monitor)	XEN
Cost per VM$/Hr	0,1
Memory Cost $/s	0,05
Storage Cost $/s	0,1
Data transfer cost $/Gb	0,1
Physical HW Units	86
Simulation duration	60 min
VM memory	512
VM image Size	10000
VM Bandwith	1000
Processor speed	DC1: 1000 Mb DC2: 700Mb DC3: 300 Mb

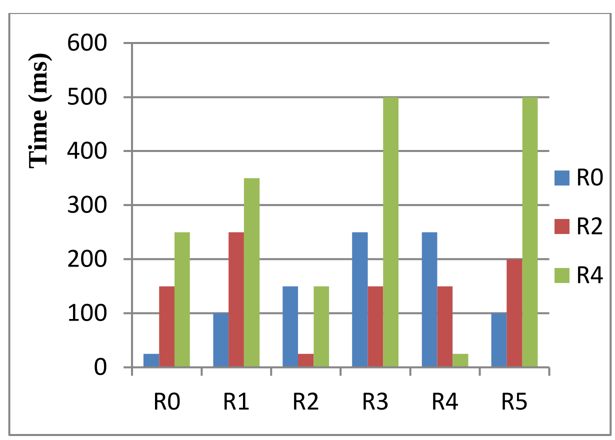


Fig. 7 Transmission delay between regions (units in milliseconds)

After performing the simulation, the results computed by cloudAnalyst are as shown in the above figure 8. We observe that the nearest the cloud resource the better the response time. In other words, by bringing the service closer to users we enhance the quality of service (response time in this case). It is an expected fact, because users experiment less effects from Internet issues (high latency, low bandwidth) when they are geographically close to the application server. That's why we propose in our solution "adapter request module" for selecting the cloud provider according to mobile users' locations. Hence, we gain in terms of the delegation's response time as well as maximize the lifetime of mobile devices.

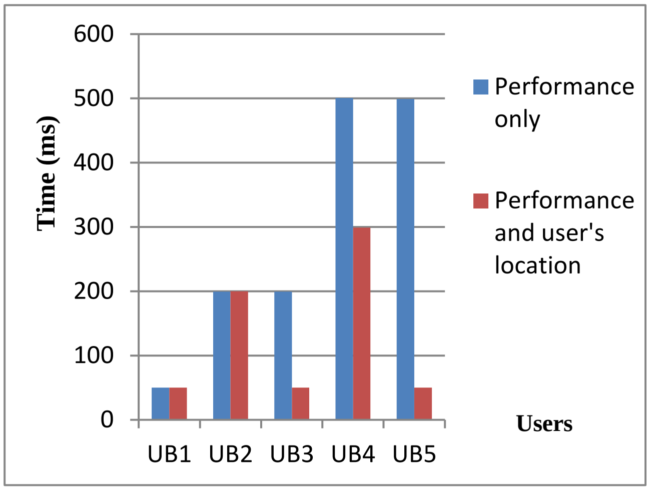


Fig. 8 User Base Hourly Response Times (Simulation duration is 60 min)

E. Scenario 2:

We have used the CloudAnalyst to evaluate the caching-analyzer based algorithm 2.By considering the transmission delay between regions (figure 9), we have taken two identically configured cloud resources to handle the user's request. One cloud resource is located in R0 and another in R2. In both cloud resources, DC1 (local service for R0) and DC2 (local service for R2), there are the same performance capabilities when they are request locally.

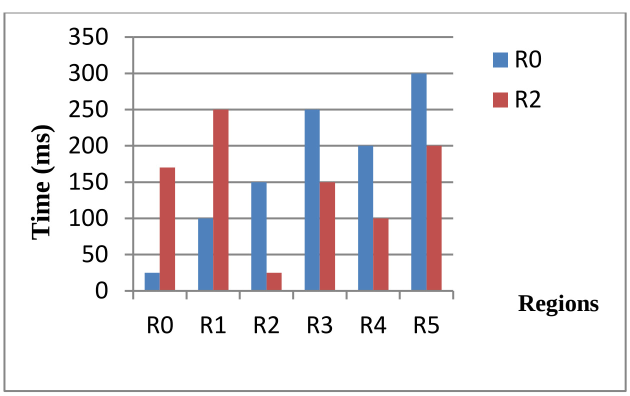


Fig. 9 Transmission delay between regions (units in milliseconds)

First case we exploit a user's service access history. In R0, the user consumes service from DC1. Then we change the user's position in several locations around the globe with the utilization of the previous selected resource DC1. However, its response time is very long when the user uses this out R0 or R1 (figure 10). That means, a service could be efficient in one moment, but fail in another moment due to the change of user's location.

Second case, the user consumes service from DC1 and changes his location from time to time around the globe. So, the proposed algorithm makes the evaluation of cloud resource DC1 whether it is suitable or not for the user's location, which may reduce response times and satisfy the user that demands to consume the suitable service in real time. If suitable response is not found, the middleware selects another resource in order to reduce the service access delay (Table 4). By using the middleware and considering the location of user (figure 10), we can observe that the proposed algorithm 2 gives better response time than exploiting the service access history of user, which focus to select the preferable service according to the user's usage without taking into account users 'movement.

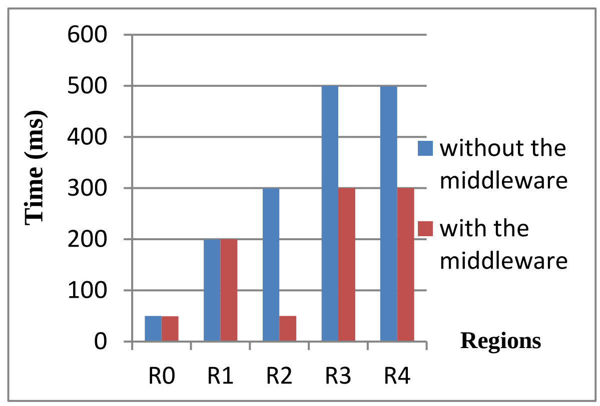


Fig. 10 Average response time

Table 4: User's movement of among regions

	R0	R1	R2	R3	R4
Without middleware	DC1	DC1	DC1	DC1	DC1
With middleware	DC1	DC1	DC2	DC2	DC2

6. Conclusion

In this paper, we introduce a new middleware which tackles how the cloud providers can exploit the context information of their mobile clients in order to provide a suitable cloud service. Moreover, our solution tries to define the new interaction with the existing cloud resources, from which mobile users demand customizable services depending on their context. As shown in the simulated results, the exploitation of mobile users' locations aids to select the nearest cloud resource and minimizes the response time of delegation.

As part of our future work, we plan to exploit the other context factors influencing the mobile cloud, such as energy and cost, to understand how users interact with these devices and define a complex set of conditions, which are utilized to optimize mobile cloud applications. Even though context aware applications in a pervasive world require understanding users' contexts in order to provide the necessary cloud services, the more user context is acquired by a system the more the user is concerned about his/her privacy and security. Thus, other future works to be considered are privacy, security, and performance analysis of the suggested architecture.

References

[1] Bangui, H., &Rakrak, S. (2014, November). Mobile Cloud Middleware: Smart Behaviour for Adapting Cloud Services. In Signal-Image Technology and Internet-Based Systems (SITIS), 2014 Tenth International Conference on (pp. 682-686). IEEE.

[2] Global mobile data traffic forecast update, 2013–2018," Cisco White paper, 2014. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html

[3] Abolfazli, S., Sanaei, Z., Gani, A., Xia, F., & Lin, W. M. (2014, December). RMCC: Restful Mobile Cloud Computing Framework for Exploiting Adjacent Service-Based Mobile Cloudlets. In Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on (pp. 793-798). IEEE.

[4] Yang, D., Sonmez, M. M., Li, Q., & Duan, Y. (2015). The power of triple contexts on customer-based brand performance—A comparative study of Baidu and Google from Chinese netizens' perspective. International Business Review, 24(1), 11-22.

[5] https://www.yandex.ru/

[6] Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: architecture, applications, and approaches. Wireless communications and mobile computing, 13(18), 1587-1611

[7] Khan, A. R., Othman, M., Madani, S. A., & Khan, S. U. (2014). A survey of mobile cloud computing application models. Communications Surveys & Tutorials, IEEE, 16(1), 393-413.

[8] Elgazzar, K., Martin, P., &Hassanein, H. S. (2013, December). Empowering mobile service provisioning through cloud assistance. In Utility and Cloud Computing (UCC), 2013 IEEE/ACM 6th International Conference on (pp. 9-16). IEEE.

[9] David, L., Vasconcelos, R., Alves, L., André, R., Baptista, G., &Endler, M. (2012, June). A communication middleware for scalable real-time mobile collaboration. In Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2012 IEEE 21st International Workshop on (pp. 54-59). IEEE.

[10] Bohez, S., De Coninck, E., Verbelen, T., Simoens, P., &Dhoedt, B. (2014, December). Enabling component-based mobile cloud computing with the AIOLOS middleware. In Proceedings of the 13th Workshop on Adaptive and Reflective Middleware (p. 2). ACM.

[11] Liu, D., & Xiao, P. (2013). User-oriented mobile filesystem middleware for mobile cloud systems. Journal of Computers, 8(9), 2209-2216.

[12] Shiraz, M., Gani, A., Khokhar, R. H., &Buyya, R. (2013). A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing. Communications Surveys & Tutorials, IEEE, 15(3), 1294-1313.

[13] Wu, Y., Fu, W., Shao, J., &TianZhou, C. (2013, July). Design of a novel cloud terminal adaptation platform based on mobile web-middleware. In Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech), 2013 First International Symposium on (pp. 1-5). IEEE.

[14] Lomotey, R. K., & Deters, R. (2013, March). Middleware-Layer for Authenticating Mobile Consumers of Amazon S3 Data*. In Cloud Engineering (IC2E), 2013 IEEE International Conference on (pp. 108-113). IEEE.

[15] Abolfazli, S., Sanaei, Z., Gani, A., Xia, F., & Lin, W. M. (2014, December). RMCC: Restful Mobile Cloud Computing Framework for Exploiting Adjacent Service-Based Mobile Cloudlets. In Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on (pp. 793-798). IEEE.

[16] Flores, H., &Srirama, S. N. (2014). Mobile cloud middleware. Journal of Systems and Software, 92, 82-94

[17] Cedillo, P., Jimenez-Gomez, J., Abrahao, S., &Insfran, E. (2015, June). Towards a Monitoring Middleware for Cloud Services. In Services Computing (SCC), 2015 IEEE International Conference on (pp. 451-458). IEEE.

[18] Go, Y., & Song, H. (2013, September). Preference-aware skyline service discovery in the mobile cloud system. In Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on (pp. 3543-3547). IEEE.

[19] Singh, M., &Dhindsa, K. S. (2013). Enhancing Interaction between Smartphones and Web services on Cloud for improved Bandwidth and latency. International Journal of Computer Science and Mobile Computing (IJCSMC), 2(4), 177-185.

[20] Fernando, N., Loke, S. W., &Rahayu, W. (2013). Mobile cloud computing: A survey. Future Generation Computer Systems, 29(1), 84-106.

[21] Chen, G., &Kotz, D. (2000). A survey of context-aware mobile computing research (Vol. 1, No. 2.1, pp. 2-1). Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College.

[22] A.K. Dey, ―Understanding and Using Context, Journal of Pervasive and Ubiquitous Computing, vol. 5(1), pp. 4–7, 2001.

[23] Chaari, T., Ejigu, D., Laforest, F., &Scuturici, V. M. (2007). A comprehensive approach to model and use context for adapting applications in pervasive environments. Journal of Systems and Software, 80(12), 1973-1992.

[24] Weiser, M. (1991). The computer for the 21st century. Scientific american, 265(3), 94-104.
[25] https://developer.foursquare.com/
[26] Bilogrevic, I., Huguenin, K., Mihaila, S., Shokri, R., &Hubaux, J. P. (2015). Predicting Users" Motivations behind Location Check-Ins and Utility Implications of Privacy Protection Mechanisms. In 22nd Network and Distributed System Security Symposium (NDSS" 15) (No. EPFL-CONF-202202).
[27] De Domenico, M., Lima, A., &Musolesi, M. (2013). Interdependence and predictability of human mobility and social interactions. Pervasive and Mobile Computing, 9(6), 798-807.
[28] Ba, H., Heinzelman, W., Janssen, C. A., & Shi, J. (2013, April). Mobile computing-A green computing resource. In Wireless Communications and Networking Conference (WCNC), 2013 IEEE (pp. 4451-4456). IEEE. DOI: 10.1109/WCNC.2013.6555295
[29] Ibrahim, S., Jin, H., Cheng, B., Cao, H., Wu, S., & Qi, L. (2009, June). CLOUDLET: towards mapreduce implementation on virtual machines. In Proceedings of the 18th ACM international symposium on High performance distributed computing (pp. 65-66). ACM.
[30] Elzeki, O. M., Reshad, M. Z., &Elsoud, M. A. (2012). Improved Max-Min Algorithm in Cloud Computing. International Journal of Computer Applications, 50(12), 22-27.
[31] Wickremasinghe, B. (2009). CloudAnalyst: A CloudSim-based tool for modelling and analysis of large scale cloud computing environments. MEDC project report, 22(6), 433-659.
[32] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., &Buyya, R. (2011). CloudSim: a toolkit for modeling and simulation of cloud computing environment and evaluation of resource provisioning algorithms. Software: Practice and Experience, 41(1), 23-50.
[33] Bangui, H., Rakrak, S., &Raghay, S. (2015, June). External sources for mobile computing: The state-of-the-art, challenges, and future research. In Cloud Technologies and Applications (CloudTech), 2015 International Conference on (pp. 1-8). IEEE.

Hind Bangui is currently a Ph.D student with the Applied Mathematics and Computer Science Laboratory of the Cadi Ayyad University (UCAM) in Morocco. Her research is mainly focused on Mobile cloud computing.

Said RAKRAK is a Professor and researcher with the Laboratory of Applied Mathematics and Computer Science, Faculty of Science and Techniques, Cadi Ayyad University, Marrakesh, Morocco.

Said Raghay is a Director of University Center Kalaa Sraghna- Cadi Ayyad University and the head of Applied Mathematics and Computer Science Laboratory at Cadi Ayyad University of Marrakech (UCAM).