International Journal of Advanced Research Methodology in Engineering & Technology,
Volume 1, Issue 2, March 2017, ISBN 978-1-63535-889-6

Location-based services in mobile cloud computing

Harish Naik B M
Assistant professor: Dept of CSE
T JOHN Institute of Technology
Bangalore, India
E-mail: harishnaik@tjohngroup.com

***Abstract*— Mobile cloud computing (MCC) has changed the concept of mobile devices from primitive gadget to full computers that accommodate work, personal and mobility needs. MCC is the most promising cloud solution for the future mobile environment. It aims to integrate mobile devices with cloud computing, and provide to mobile users an online access to unlimited cloud resources. This paper introduce a middleware that provides an intelligent behavior for selecting and adapting cloud services according to the current user's context. Implementation of context-aware algorithm aiming to exploit location and preference cost of mobile user to select the adequate cloud resource.**

Keywords*— cloud computing; mobile computing; mobile cloud computing; middleware.

I. INTRODUCTION

Despite the significant improvements of mobile environments, mobile devices still encounter resource scarcity such as limited computing power, memory, and battery life, among others. To overcome this resource restriction, it has been suggested to exploit cloud computing resources via the wireless network.

Thanks to the advancement in cloud computing, mobile devices can take the benefits of cloud infrastructures such as large-scale computing power with elastic, virtualization, and pay-as-you-go notion. As a result, mobile client exploits cloud infrastructures in order to overcome their traditional limitations. Therefore, the recent collaboration, called mobile cloud computing, is a new concept that aims to improve mobile services and remove the limitations of mobile devices. For example, M-Health service is typically hosted on cloud servers accessed via communication networks. Cloud servers collect and process all the patients' service data. Hence, M-Health service facilitates efficient patient treatment for medical consultation by sharing personal health information among healthcare cloud providers.

The collaboration between cloud computing and mobile computing will change the way we consume remote services. So, mobile devices access to the cloud server to consume the unlimited cloud resources at any time, even though they are not similar to the traditional terminal devices. Since, they have sensing capabilities to gather contextual information of mobile user environment. The most cited definition of context is [1]: "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". Therefore, mobile contextual information can aid the provider to deliver adoptive, efficient, and context-aware services for users which demand to consume suitable service according to their actual situation. So, it is necessary to benefit from such information for mobile cloud services' improvement.

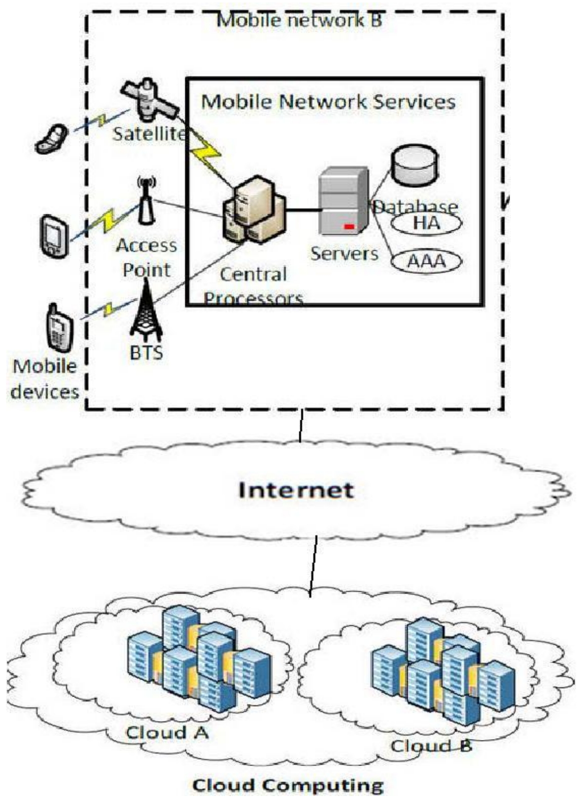


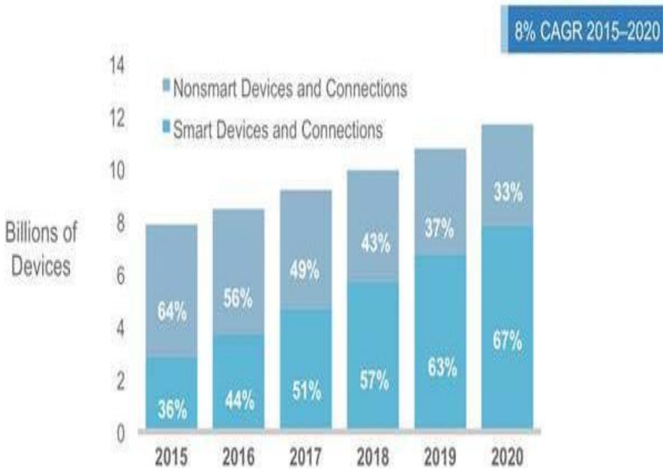
Fig. 1. Mobile cloud computing architecture

The goal of this paper is to introduce a middleware that gathers context data from mobile device, such as location. Then it selects a service according to the available cloud services and users' requirements. Further, it exploits the user's preferable cost, which offers more accurate information to better select the service provider.

The remainder of the paper is structured as follows. In section II presents some important related work. Section III, presents the middleware architecture with the description of the interaction between its modules, section IV, conducting a primary experiment to evaluate the performance of the proposed middleware. Section V concludes the paper and the points to future work.

II. RELATED WORK

According to [3], the sharing smart devices and connections will increase from 36 percent in 2015 to more than two-thirds, at 67 percent, by 2020 (figure 2). So, the middleware has attracted a lot of attentions from researchers in both academic and industrial communities, in order to enhance the interaction between mobile devices and cloud providers. Similarly, it aims to ensure that mobile clients make the best use of cloud computing resources in order to benefit from all the massive advantages of the cloud [13].



Percentages refer to device and connections share.

Fig. 2. Global Growth of Smart Mobile Devices and Connections

Mobile cloud applications represent one of the newest developments in cloud computing advancement. Mobile Health Net [4] is a middleware for mobile social networks, which is used as a software infrastructure for the creation of new services and applications focusing on the healthcare domain. As for the middleware called Device Nimbus [5], it is considered as another solution for continuous health monitoring. Another initiative presented in [6], namely WellPhone, collects physiological data from different devices.

Recently, many research works were interested in enhancing the collaboration between cloud services and mobile environment. In [6], the authors propose an adaptation platform to bridge between the mobile terminals and cloud. They present the technique framework of web-middleware that acts as a standard interface for the mobile middleware and the cloud.

Similarly, in [7], the authors present framework, called mobile cloud middleware (MCM) that addresses the issues of interoperability across multiple clouds, transparent delegation and asynchronous execution of mobile tasks that require resource-intensive processing, and dynamic allocation of cloud infrastructure. In order to minimize the number of of loading time needed in a mobile cloud application, MCM can develop a customer services based on service composition. In fact, the previous middleware architectures benefit from all the advantages of the cloud computing; however, they do not take advantage of the mobile devices by exploiting their current context.

Meanwhile, mobile cloud computing is born to bring new types of services to mobile users and utilize cloud computing to provide ubiquitous mobile service access. A very interesting research has been conducted in [8], where the authors provide a complete survey on the field of mobile cloud computing on its state of art applications, research challenges, opportunities and future research directions. Such as context-awareness which becomes the subject of recent fields of research in pervasive computing, where the applications change and adapt the behavior according to their user and his situation.

Actually, there are several definitions for context such as [9], where the authors propose a definition of the 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."

With the obvious advantages of context-aware, it is important to benefit from such piece of information for mobile cloud services' improvement. Further, the mobile cloud computing environment is not only powerful servers in remote data centers and personal devices, but it is a combination of mobile computing and cloud computing. So, it has also the advantage of mobile computing (i.e. users' preferences, location, time, etc.).

On the other hand, the study in [2] describes how the triple contexts (country, industry and firm) affect customers' views of corporate brands. Besides, it compares between a global and local service (Google as a global brand and Baidu as a dominant Chinese brand). So, context can greatly impact the cloud service. For example, Yandex[10] offers several services and performs very well in Russia than Google. However, its performance is very poor outside Russia because it is built to understand local culture and Russian netizens. Otherwise, a cloud resource can be performed or satisfy mobile user in a certain moment. In order to create ubiquitous mobile cloud service, it is necessary to investigate the location of mobile users. Yet, the goal of our solution is to acquire and utilize the context of mobile devices (such as location) in order to provide services that are appropriate to the mobile client.

III. MIDDLEWARE ARCHITECTURE:

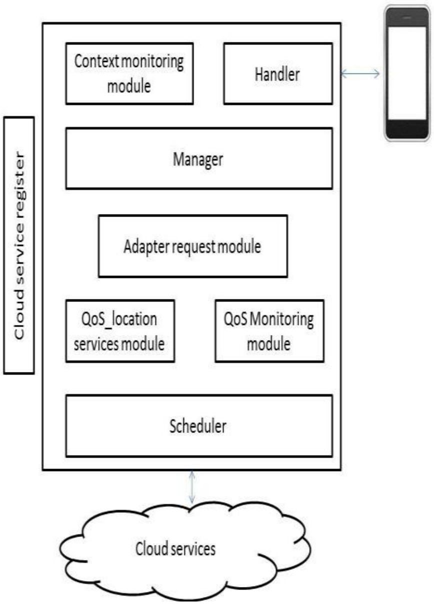


Fig. 3. Middleware architecture

The proposed solution is an adaptive delegation mechanism that aims to perform interaction between the mobile device and cloud. Acting as an intermediary, the middleware intercepts service discovery requests from applications while monitoring mobile device's context such as battery consumption, CPU usage, network bandwidth, data from sensors, And so forth. Depending on this contextual information, the middleware tries to adapt each service request.

Table 1 Notations

Notations	Explanations
T_i	Task
Ri	Resources
Cij	Expected completion time
Eij	Estimated execution time
r_i	Time to become ready to execute a task
Tt	Transmission time
Td	Delivery time

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

Fig. 4. The Traditional Max-Min Algorithm

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 adapter request module that extends the traditional Max-Min Algorithm [11]. This algorithm selects the task with the maximum completion time, and assigns it to the faster resource in order to achieve minimum execution time.

Algorithm 1

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

The adapter request module handles the service requests according to the current context of the mobile device provided by the context monitoring module and the QoS level. The QoS level is selected from the QoS-location services module which contains service lists on the cloud resources and their QoSs in different locations, this module will be updated by considering mobile users mobility. So, the adapter request module exploits the users' location to select the nearest cloud resource. Next, it uses the estimated completion time for each cloud resource to select the adequate QoS level. In case mobile user specifies cost, the adapter selects the service according to the user's cost preference; we present the algorithm (1) in this section that is applied by the middleware.

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 will advertise 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 will send the request to the provider in the cloud.

Once the middleware receives the response, the manager sends back the result to the handler and notifies the mobile client by using the asynchronous notification.

1. SIMULATION AND RESULTS:

Cloud Analyst [14], which is graphical cloud simulation software on the basis of CloudSim [15], to evaluate algorithm 1

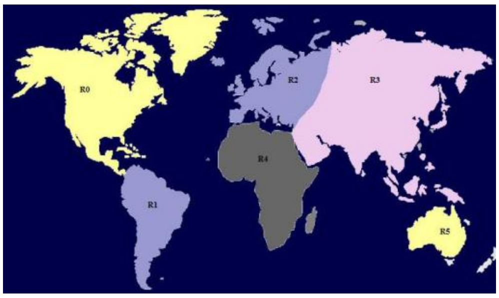


Fig. 5. Regions in CloudAnalyst

In cloud Analyst, the world is divided into 6 regions (figure 5). In order to create such environment, 2 data centers (DC) are created with 70 and 100 virtual machines. Table II shows data center configuration used in simulation. Then 5 user base (UB) is created from all the regions of the world. Next, we integrate algorithm 1 in the broker which determines the region of user base and then asks for the region proximity list with respect to the region of the sender of the requests. The list is ordered according to the user's location (nearest to fairest).

Table II. Data center configuration

Parameters	
Name	DC1(region 0) , DC2(region 2)
Data Center Architecture	X86
Data Center OS	Linux
VMM	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

Results:

The Response time for each user base is calculated by the cloud Analyst. Yet, the simulation result computed by the Cloud Analyst is shown in Table III. After performing the simulation the result computed by cloud Analyst, we can observe that as data center near user as it gives better response time. The primary results show that bringing the service closer to users improves the quality of service (response time in this case). It is an expected effect, because users experiment less effects from Internet issues (high latency, low bandwidth) when they are geographically close to the application server. That's why solution module is proposed for selecting the cloud provider according to location of mobile user in order to minimize the response time of delegation while maximizing lifetime of mobile devices.

Table III. Response Time by Region

User base	Avg (ms)	Min (ms)	Max (ms)
UB1	51,57	43,46	65,76
UB2	203,55	171,34	247,24
UB3	51,20	42,01	57,06
UB4	304,50	258,94	343,57
UB5	298,75	251,04	345,97

IV. CONCLUSION:

Mobile devices hold multiple sensors, like GPS, which can collect a set of information about the mobile user's environment. So, deploying context of these devices can optimize applications, and aid providers to better understanding of mobile users' needs

The primary objective of this research is to introduce a middleware, which exploits the mobile devices' location and users' cost preference to select the adequate cloud resources. In the future, planning to develop a prototype for the proposed middleware, and evaluate the performance analysis of the suggested architecture.

REFERENCES

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

[2] 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.

[3] 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

[4] Teles, Ariel, et al. "Mobilehealthnet: A middleware for mobile social networks in m-health." *Proceedings of the 3rd International Conference on Wireless Mobile Communication and Healthcare, MobiHealth*. Vol. 12. 2012.

[5] Oliveira, E. A., Koch, F., Kirley, M., & dos Passos Barros, C. V. G. (2015). Towards a Middleware for Context-Aware Health Monitoring. In Advances in Social Computing and Multi agent Systems (pp. 19-30). Springer International Publishing.

[6] Moser, Louise E., and P. M. Melliar-Smith. "Personal Health Monitoring Using a Smartphone." *2015 IEEE International Conference on Mobile Services*. IEEE, 2015.

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

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

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.