Cloud-Enabled Hybrid Architecture For In-Class Interactive Learning Using Mobile Device

Mohammad Muztaba Fuad
Dept. of Computer Science
Winston-Salem State University
Winston-Salem, NC 27110, USA
fuadmo@wssu.edu

Debzani Deb
Dept. of Computer Science
Winston-Salem State University
Winston-Salem, NC 27110, USA
debd@wssu.edu

Abstract— **Mobile learning environments have the benefit of facilitating real time student learning and assessment. However, most of such learning environments only support static or traditional learning activities. In STEM disciplines, we need more active and engaging activities and mobile learning environments should be able to support such dynamic activities. By designing such learning environments to run completely on the cloud will limit its extensibility and will not accommodate interactive activities developed by anyone other than the developer of such learning environment. Instead, we argue that by incorporating cloud services in a traditional software architecture will allow the flexibility to develop and deploy interactive problem solving activities along with versatility that cloud computing brings. This paper presents Mobile Response System (MRS) that facilitates in-class interactive problem solving using mobile devices. MRS uses cloud services in the infrastructure to minimize instructor's workload, gives students transparent access, and makes the system failsafe alongside providing extensibility to any discipline.**

Keywords—interactive problem solving; mobile learning; pervasive learning; cloud services.

I. Introduction

In recent years, the technological advances in mobile devices allowed its users to use it in domains for which it was not intended for. One such domain is education, where the steady growth of mobile device usage has nearly eliminated the boundary between formal and informal learning. Inside and outside the classroom, a new trend has emerged, where learning is technology enhanced, specially through the use of mobile devices. There are growing indications [1] that, such technology enhanced learning improves the quality of both teaching and learning. However, a big limitations is that most of such mobile device usage in the classroom tend offer content delivery, multiple-choice quiz taking, voting etc., which are static in nature. For active learning, we need more engaging and interactive problem solving using those devices and the feature set of current mobile devices is powerful enough to support such interactivity. This paper presents a mobile learning environment, called Mobile Response System (MRS) [2] to help students solve interactive exercises (IE) in their mobile devices to improve their class engagement, active learning and problem solving skills. Currently, cloud-based learning solutions are introduced extensively in academia, some of which also use mobile devices as delivery endpoints. Since cloud computing environments provides 24/7 service seamlessly, there is a great potential to develop learning environments which are completly hosted on the clouds. However, such development scenario is limited by end-user's choice of development environment, programming languages, extensibility and interoperability with existing learning management systems. MRS provides a cloud-enabled hybrid software architecture with three major features. Firstly, it supports interactive problem solving using mobile devices where the interactive activity apps are independent of the software architecture and therefore can be developed by others. Secondly, the architecture uses cloud-computing capabilities to provide a semaless and failsafe learning environment. Finally, It uses existing programming languages and development environments to use and to extend the system in any discipline. Having these charactarestics makes MRS unique and shows a neat way to harness the power of the cloud while enhancing classroom education.

II. Related Works

Different studies [3]-[4] discussed the limitations of traditional pedagogical approaches in improving student learning, specifically the transfer of critical knowledge to students. These studies also shown that more interactive teaching and learning strategy is necessary to make learning more productive. In the last couple of years, several studies [5]-[6] have found the benefits of using mobile devices in classrooms. There exist several commercial products [7]-[8] for different mobile platforms that create more active, student-centered learning environment while supporting various effective pedagogies. However, there are two basic problems with these systems. Firstly, and most importantly, such systems only support multiple-choice or true/false type of questions and unable to support multi-step problem solving activities. Specially in STEM courses, students need to develop skills such as synthesizing a problem and critical thinking by using such problems in a hands-on approach. Secondly, related mobile-based learning environments are either completely mobile-based [5]-[6] or they are completely cloud based [9]-[10]. Although both approaches have their own advantages and disadvantages [1], [11]-[12], by combining the extensibility and robustness of desktop-centric approach with the flexibility and 24/7 availability of cloud-based approach, it is possible to introduce a hybrid system that will provide better fault tolerance, seamless operation and will benefit both instructor and students during instruction. Our arguments for the hybrid design are as follows:

- A generic front-end with application programmer interface (API) services allows extensibility without the need for addressing cloud operations, which are hidden by the architecture. This will allow third-party developers to create new IE apps without thinking about programming cloud services.
- Having a system-level integration of cloud services in the architecture makes cloud IaaS transparent to the end user and allow for seamless operation. This will reduce application development time, as developers do not have to worry about cloud operations.
- The hybrid architecture allows the system to be more portable, available and safer to use because of the recoverability of all crucial and sensitive student interaction information. This will allow instructor and students to use the system without worrying about possible loss and corruption of student interaction and also will allow for verification in case of a system crash.

III. MRS System and Its Architecture

MRS is a software environment that facilitates in-class problem solving exercises and their real-time assessment utilizing mobile devices. MRS allows the instructor to dynamically prompt the students with carefully designed exercise, synchronized with the lecture material, in their mobile devices. students are able to actively interact with the exercise while recognizing the effect of their interactions visually at different stages of the algorithm and send their solutions back to the faculty computer. MRS then facilitates grading of the exercises. Since the delivery, time keeping and grading of the exercise apps are automated in this technology-enhanced learning environment, instructor could use them more frequently in the classroom and learners could actively participate in more exercises where they can receive feedback. MRS also supports in-class anonymous communication, where students can send anonymous question/feedback to the instructor and additionally can vote on existing pool of questions that instructor may choose to review and answer at the end of the class. Another important aspect of MRS is the integration of instant data visualization that analyzes vast amount of student performance, interactions and mobile device usage data (i.e. button clicks, time spent, navigation behavior etc.) in order to summarize and interpret student performance, mental model, and attitudes during problem-solving.

A. Software Architecture

The MRS software is designed as a three-tier architecture as shown in Fig. 1. The instructor computer runs the server component of the software and the client component executes in student's mobile device. Java is used to develop the server and currently the client component supports Android mobile devices. A cloud IaaS is used as the middle tier to facilitate seamless operation. The cloud middle-tier acts as a broker between the clients and the server. Having these different tiers with associated functionality provides the developer and the user flexibility to use and to extend the software in a domain independent manner. If the software is developed solely on the cloud or solely as a client-server architecture, it would lack the proposed flexibility and extensibility. Both the server and the client provides API interface for users to develop their own IE apps and corresponding grading modules to be used within MRS environment.

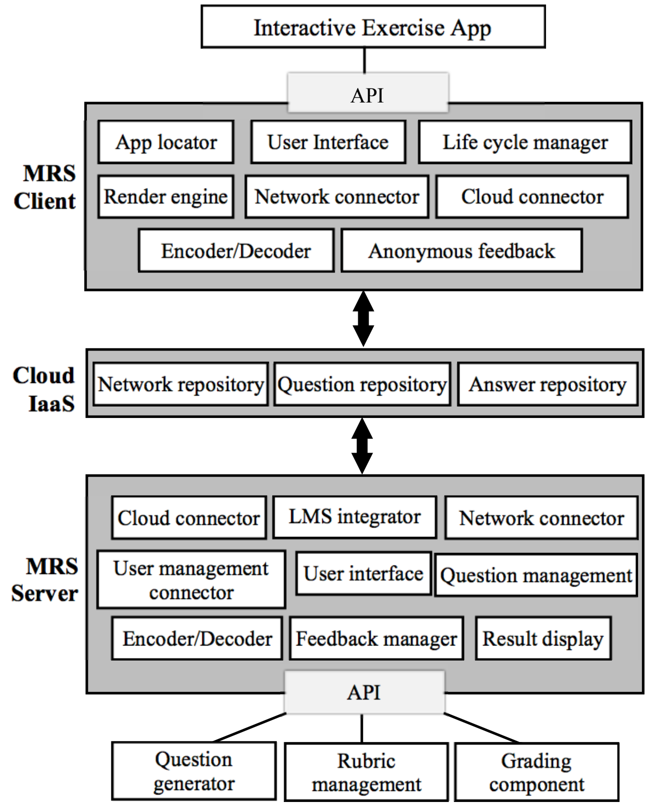

Figure 1. MRS Software Architecture.

The server is the major software entity in the system and the client contains platform specific components to facilitate interactive problem solving. The cloud middle-tier acts as a repository for different databases for the system. These databases are maintained as redundant backup for all operational data, such as client and server address, question being broadcasted, student responses etc. Two of the biggest challenge of cloud computing is security and privacy, which are exacerbated in a student learning environment dealing with student's sensitive data. To address that, MRS uses it's across the system encoding and decoding technique to encrypt any data transferred between the tiers and also anything stored in the cloud.

B. MRS Client-Server Interaction

To have a seamless operation, the client and the server needs to interact on the background to facilitate student login, question broadcasting, result gathering and feedback/question management.

Fig. 2 shows sequence diagram of major components for three major interactions. At startup, once the instructor starts the server, it registers itself with a cloud-based broker and saves contact information for clients to use. Having a cloud-based architecture allows the user to seamlessly use the system without the need for system installation and setup. The cloud services are completely hidden from user and there is no setting up or maintenance of that from user's part. This study is using a freely available cloud service [13] as cloud IaaS, which support enough clients to facilitate most class size. On the client side, MRS simultaneously broadcast a ping to locate the server and also contacts the broker, which has a static address. From either or both, the client then collects the server information. With that information, it does a quick handshake with the server to check the validity of that address and whether the server is indeed running. Once successfully hand shaken, students are asked for their login credentials. After successfully verified by the server, students are allowed into the system. When the instructor broadcasts a question, the server sends a message with an encoded version of the question to the client and also saves a copy on the cloud. After the students submit their answer, the client app sends the answer back to the server using another encoded message and also save a copy in the cloud database. In terms of submitting a new feedback/question, the client app first receives the list of existing feedback/question from the server and sends any new or voted feedback/question back to the server. All of these interactions are made using background threads to keep the system responsive and the messages use UDP packets to make communications non-blocking.

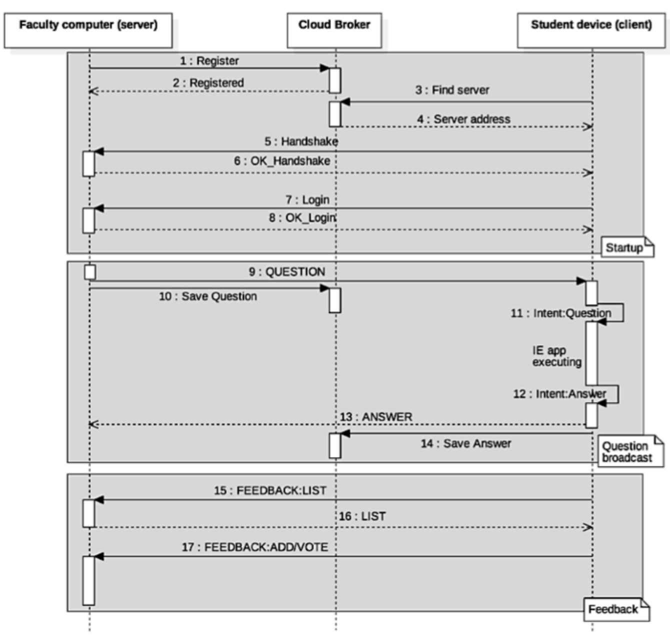

Figure 2. Sequence diagram of major system interactions.

C. Interactive Exercise(IE) Structure

The most important aspect of the MRS software is the facilitation of interactive problem solving. An interactive problem solving activity is defined as an Interactive Exercise (IE) with corresponding grading components and associated rubrics. In an interactive exercise, students develop the answer of a problem by following a set of steps under a particular algorithm or process. In each step, students make key choices that have impact on the next step of the interaction. During these interaction steps, students can go back and forth and see the impact of their different choices. Interactive exercises can be offered as solving a whole problem from bottom-up or top-down or solving certain steps of a particular problem in order to give students different perspective on the problem and to rigorously assess their problem-solving skills.

An IE definition is stored as an XML file and contains exercise parameters and grading related information. Each IE has two major parts. One defines the problem itself with individual problem components and other allows grading the student answers with different rubric and grading components. MRS provides appropriate API functionality to manipulate the IE definition structure and to encode and to decode it for transfer and rendering.

D. Extensibility

It is impractical to build IE apps for each possible problem type within the client. This will not only make the client overly bulky to run in mobile platform, but will also make it domain centric. To overcome this, the IE activity (or the app) that facilitates interactive exercise is completely separated from the application logic of the MRS client software. The below listed requirements and processes need to be followed when developing user defined IE and corresponding server components:

- A server-side program needs to be developed to manipulate the IE and to grade its answer. The requirements are as follows:
  - The program should have three methods, *define*, *edit* and *grade* so that MRS can dynamically call them for corresponding functionality using reflection. The grade method is preceded by a *setUser(..)* method to allow user application to send any feedback and grade information to the students using e-mail or other methods of communication.
  - The program should be in a package with the same name as the package of the client-side IE app and the corresponding question type.
- Once the IE is defined and saved in the storage as a XML representation, it can be imported to MRS by using the MRS GUI, which uses ***MRS.import*** to read the IE definition on the file. One file can have multiple types of IEs in it with unique question numbers.
- Instructor uses the broadcast button in MRS, which uses the ***MRS.encode*** method to broadcast it to all available clients.
- Once the MRS client receives the encoded definition, it finds the corresponding IE app and invokes it with the encoded definition.

- Once the IE app is invoked, it needs to get the encoded question and then uses the ***MRS.decode*** method to convert it to the workable definition format.
- After student submit an answer, the IE app terminates and returns the student response to MRS. Once MRS client receives that Intent, it forwards the contained response back to the server using ***MRS.encode.***
- MRS server locates the server side program to process the student responses and invokes the *grade* method using Java reflection in order to produce grade statistics.

Following the above sequence of steps, it is possible to integrate and execute any IE app developed by third party developers into the MRS software. Third party developer can also decide on the level of interactivity they want to incorporate in their apps and the way to tackle activity life cycle for their app.

IV. EVALUATION

MRS was rigorously evaluated to measure its effectiveness in student learning, engagement and student satisfaction. We also evaluated software engineering quality matrices for MRS software and its performance during runtime. In this paper, we investigate and measure the effect of embedding cloud services in the architecture. To measure the response time, we investigate how the client and server components interact with each other in different situations (Section III.B) and the effect of having a cloud broaker as the middle-tier of the architecture. To measure MRS response time at startup, we measure the timing data that different components utilize during startup (Fig. 3(a)). Although, the interaction with the cloud broaker is relatively high compared to the other components during startup, the overhead is insignificant (less than 10 milli-sec) and is worthwhile considering the flexibility and versatility that it provides.

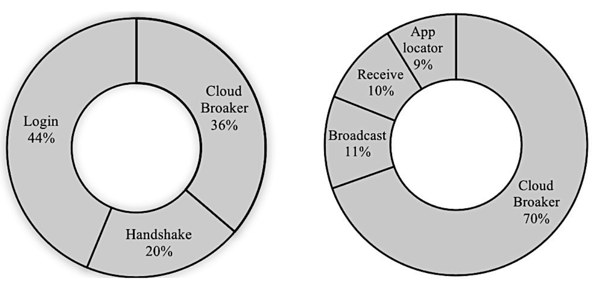

Figure 3. MRS response time.

When a question is broadcasted and students answers are received, multiple components contribute to that response time. On the server side, the question needs to be encoded, and then send to each of the clients (broadcast) and simultaneously need to be sent to the cloud broker. On the client side, the question is decoded and the corresponding IE app is located and invoked with the given question (app locator). Once IE app is done executing, the answer is sent back to the server (receive) and the cloud broker. Fig. 3(b) shows the breakdown of this timings. The time for the cloud broker is the combination of the time of saving the question and its answer in the cloud. Although the overhead seems significant compared to the other components, the actual additional time is less than 45 milli-seconds. From Fig. 3, it is evident that incorporating the cloud-tier add substantial amount of overhead, however, since at any time MRS is only working with one question, the system is scalable with respect to the number of students and since it is all done concurrently in the background, there is no impact of that in user's impression of system performance.

V. CONCLUSIONS

This paper illustrates the incorporation of cloud services in a mobile-based learning environment that offers interactive problem solving and describes the benefits of such software architecture. MRS provides a responsive environment where students can solve their problems in an interactive way and communicate their solutions with the instructor and the instructor can communicate questions, answers, feedbacks and student performance data with the students. By incorporating cloud services in such software architecture, MRS makes it accessible, extensible, failsafe and productive for instructor to use and to provide student with a positive in-class learning experience.

ACKNOWLEDGMENT

This research was supported by National Science Foundation grant #1332531.

REFERENCES

[1] Sotsenko, A., Zbick, J., Jansen, M., & Milrad, M. (2016). Flexible and Contextualized Cloud Applications for Mobile Learning Scenarios. In *Mobile, Ubiquitous, and Pervasive Learning* (pp. 167-192). Springer International Publishing.

[2] Mobile Response System (MRS), http://compsci.wssu.edu/MRS/.

[3] Bruff, D. (2012). Classroom Response System Bibliography, Vanderbilt Center for Teaching.

[4] Costa, C. E. Ojala T. and Korhonen, J. (2008). Mobile lecture interaction: Making technology and learning click. Presented at the International Association for Development of the Information Society Conference on Mobile Learning, Algarve, Portugal.

[5] Wong L. H. and Looi, C. K. (2011). What seams do we remove in mobile assisted seamless learning? A critical review of the literature, Computers and Education, 57(4), 2364–2381.

[6] Young, R. J. (2008). Mobile College App: Turning iPhones Into 'Super-Clickers' for Classroom Feedback, *Chronicle of higher education*.

[7] Top Hat Monocle, 2016, https://www.tophatmonocle.com.

[8] Socrative, 2016, http://www.socrative.com.

[9] Boyinbode, O., & Akintade, F. (2015). A Cloud Based Mobile Learning Interface. In *Proceedings of the World Congress on Engineering and Computer Science* (Vol. 1).

[10] AO, A. 2014, On Mobile Cloud Computing in a Mobile Learning System, Journal of Information Eng. and Applications, 4(5), 82-87.

[11] Velev, D. G. (2014). Challenges and opportunities of cloud-based mobile learning. *International Journal of Information and Education Technology*, *4*(1), 49.

[12] Sarrab, M., & Elbasir, M. Mobile Learning (M-learning) through Cloud Computing: Benefits and Challenges.

[13] DropBox developer platform, https://www.dropbox.com/developers.