Utilization of Cloud Computing for E-learning During Campus-wide Failure Situation

Muhammad WANNOUS[†1†2†3] Takayuki NAGAI[†2]
Hiroshi NAKANO[†2]

Teaching staff faced a campus-wide failure/down situation when most of private universities had to abandon their campuses and move to safer places due to the ongoing fighting in Syria. In the case of the first author, the safer place was not equipped properly with the necessary facilities for enabling the delivery of course contents over the Internet. Further more, university administration did not have the well to invest in establishing an e-learning system in the new place due to the uncertainty of continuity. The only good thing was that Internet was still working in almost all safe areas. So the teaching staff (including the first author of this contribution) thought of using the free storage service offered by a number of providers on the WEB for hosting course contents. However it did not fulfill all necessary requirements; therefore, we started working on a new system utilizing "Cloud Computing" for managing courses.

1. Introduction

Around Jan 2013 and due to the increasing insecurity and the intensive fight on the access routes, several private universities had to abandon their campuses and move to safer places inside the capital city Damascus in Syria. The time window available to complete this huge step did not exceed days; therefore, the campuses were shut down and the existing equipment was left behind.

Several members of the teaching staff (including the first author of this contribution) with one private university (IUST) were concerned about the way to deliver the materials of the courses assigned to them to the students. Those members used the on-campus e-learning system for that purpose and the new place was not equipped with such system. Further more, university administration did not have the well to invest in establishing one in the new place due to the high cost and the uncertainty of continuity.

At the beginning and to compensate the lost time of the semester, we had to distribute course materials in person via storage media (flash drive, shared PC) or wireless transfer (Bluetooth). This approach proved not to be practical for long-term use since contents are changing regularly and not all students were able to attend.

With the new semester approaching, the teaching staff decided to find a better solution for the problem. Several discussions were held to see what we needed and what we had. Fortunately, Internet connection was still available in various access mediums (mobile-based and land-line-based) around the country, so we decided to explore the possibility of using a hosting service for storing our materials as a start. Surfing the WEB for such service led us to a number of providers offering Cloud-based storage and web-hosting services (for free in some cases). The decision was made to explore the potentials behind using these services as part of the educational system.

The rest of this contribution demonstrates the approach based on "Cloud Computing" to deliver course contents.

Section 2 is a brief introduction to "Cloud Computing" and the services delivered within it, and section 3 demonstrates the initial solution we implemented to distribute course contents. This section also demonstrates the difficulties we faced in the initial approach and the feedback we received. In section 4 we introduce the new approach to achieve the goals of this research.

2. Cloud Computing

The word "Cloud" usually refers to the devices and connections forming the Internet, whereas "Could Computing" refers to "a computing environment where one party can outsource computing resources to another party via the Internet" [1]. Such resources can be networks, servers, storage applications and services.

In a "Cloud Computing" environment several layers of services are defined for the purposes of classification and simplification as illustrated in Figure 1. Each layer in this figure is intended for specific type of service to deliver within specific resources and management.

- The *Application* layer provides "*Software as a Service*" via the Internet. The users will be able to use software packages installed centrally without the need to have them on their machines. Google Apps [3] is a widely used example of SaaS.
- The *Platform* layer provides "*Platform as a Service*" via the Internet. Typical users of this service are developers and web-based service providers. Users will be able to have an integrated environment (hardware and software) for developing and deploying applications. Google Apps Engine [4] and Microsoft Azure [5] are examples of PaaS.
- The *Infrastructure* layer provides "*Infrastructure as a Service*" via the Internet. The users will be able to use hardware resources installed centrally. In this layer the user of the service is responsible for installing the necessary software. AmazonEC2 [6] is a widely used example of IaaS.

†1 Japan Society for Promotion of Science, Kumamoto University
†2 Center for Management of Information Technologies, Kumamoto University
†3 International University for Science and Technology (IUST), Syria

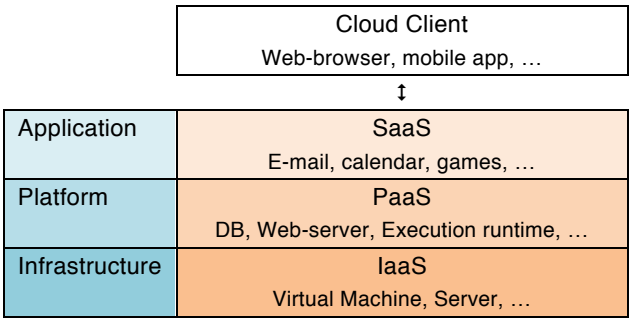

Figure 1: Cloud Computing Layers (reproduction of [2])

In some cases the term "Cloud Technologies" is used to reflect the same meaning as "Cloud Computing", but we think that the former one should be used in referring to the technologies implemented for building the services in a "Cloud Computing" environment.

3. Initial Solution (*SaaS*)

As mentioned in the introduction, right after university campus was closed we distributed course materials in person by the available media (flash memory, Bluetooth connection, common computer…etc.) But this proved not to be practical because several students were unable to attend even in the new place. It was not possible to package all contents in one file and distribute it via e-mail because we were unable to decide to which extent we were able to deliver during the time remaining.

After managing to finish the semester in which we faced the drastic change in the university location, we had a short window of time before the next semester starts. During this period, the teaching staff members hold discussions about a new approach to distribute course materials.

Since the administration of the university was not willing to spend more money on setting up a new *Learning Management System* due to many objective reasons, the new approach suggested the use of "Cloud Computing" services available for free on the Internet. This approach is reasonable and valid since Internet connection is still available in different mediums (mobile-based and landline-based) around the country.

As a start, we decided to use Google Drive [3] as a central storage for the contents of our courses. Google Drive is considered one implementation of *Software as a Service IaaS* layer. Alternatives to this selection are many, but the extra storage limit available from Google (at the time of selection) and having already accounts with it made us go for this choice.

Instructors participating in this approach created a separate folder for each course and shared it via a link to its contents with the students attending it. The folder was shared in a way that grants access only to the group of people who had the link. Using this link, students were able to access and download files from the course folder but they were not allowed to change its contents (delete, rename, change permissions) because this might cause lose of files by mistake. Instructors uploaded lecture files, assignments, laboratory activities, and demonstration contents to the folder while the semester advanced. Figure 2 shows the contents of a folder belonging to *Digital Signal Processing (DSP)* course at the end of the semester.

3.1 Pros and Cons

At this stage, we were unable to quantitatively collect feedback from the student, so we will show the results in general. Feedback from the students indicated that they were able to obtain the course materials without difficulty. Instructors indicated that distributing materials through one folder for each course was much easier than doing it in person. It was also noted that the storage limit exceeded the needs of all participants.

However, both instructors and students detected several issues. First, because access with full permissions to the course folder was not granted to students in order to prevent the deletion of contents by mistake, they could not upload their reports to course folder. Alternatively, students submitted reports manually in person.

Second, the instructor solely was responsible for the administration of the course folder and no automatic process was available for scheduling events, notification, authentication…etc. This did not ease the load of the instructor very much.

Third, it was not possible for the instructor to track the activity on the course folder, which can give an idea about what the students are doing.

Forth, it was not possible to connect the course folder to any external contents (such as laboratory activities) that might help students with their study.

The results indicated that the initial solution was not a complete failure, and that this approach can be more satisfying if we improve it by adding more components. For this reason, we decided to develop an application to manage the course folder and serve the necessary features. But rather than developing a stand-alone application we figured out that Google Apps Engine [4] offers hosting web-based applications for free. The hosting environment provides the functions of a *Platform as a Service (Paas)* layer and therefore it includes all necessary modules for deploying a web-application.

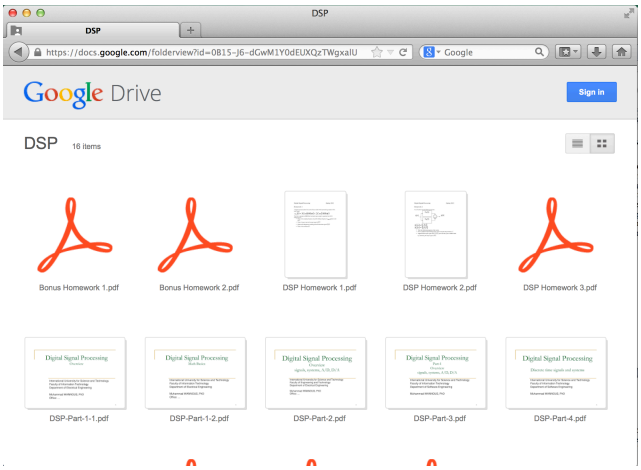

Figure 2: Course Contents distributed via Google Drive

4. Perspective Solution (*PaaS* + *SaaS*)

Several issues need to be addressed in the perspective system like activity tracking, enabling upload of reports, scheduling, course information, linking with external resources…etc. Priority was given to extracting course information, enabling upload of reports, and activity tracking because they are more important to instructors and students than the others.

4.1 Google Apps Engine

Google Apps Engine provides an integrated environment that enables deploying a number of web-application types (Java, PHP…). Currently, each Google Account has the possibility to have up to 8 applications (projects), and every application will also be able use Google specific APIs to communicate with other *SaaS* services provided by Google (Drive, Calendar, Mail…) plus other *PaaS* services (like Datastore). Google Apps Engine applications are usually publically accessed via the link "<*name*>.appspot.com" where <*name*> is the name of the project.

For an application to get access to the *SaaS* services APIs the application owner should create specific credentials for authentication purpose. In the case of a web-application the credentials are a pair of an e-mail address and a key-file that can be downloaded one time. The credentials become a part of the application and should be used in the interaction with the *SaaS* services APIs. The e-mail address in this case will have the same account privileges as a regular Google Account.

4.2 Course Management System *CMS-mini*

In Google Apps Engine we decided to create our new system (named *Course Management System CMS-mini*) intended for managing course contents and users. At the first stage we want *CMS-mini* to be able to:

- Import basic course information and course-instructor information in xml format complying with IMS Enterprise Specifications [7]. This format is widely used for exchanging course and user information and several available LMSs support it. We thought that *CMS-mini* could be used in parallel with other LMS as a back up and there for it should be able to exchange course data in standard format. In our implantation of the standard, we use the `<extension>` tag which is defined in IMS Enterprise Specifications to include information about course details
- Import detailed information about the course and the members in xml format. The xml data is referred to inside `<extension>` tags.
- Import course contents (lectures, assignments, support materials…) and make them available for download by the course members in timely manner. The instructor through a shared folder in his/her Google Drive storage will provide course contents.
- Enable course member to upload reports within the time limits defined by the instructor.
- Import scheduling information in xml format. This information will be used for deciding when certain contents should be available and defining time limits for reports and assignment.
- Import in xml format (or via the web-interface) and deliver (via the web-interface) announcements and important information.

Initial organization of *CMS-mini* is demonstrated in Figure 3. As shown in the diagram, *CMS-mini* uses the APIs provided by Google to communicate with the desired services (Drive, Datastore…). The application itself is assigned full storage on Google Drive and uses it to store the files belonging to the course (course contents which are uploaded by the instructor(s) and the reports of the students). On the other hand, the application imports the course and members information from the Google Drive that belongs to the instructor (through a shared solder). Each instructor uses the course folder stated in the previous approach to pass the information in xml format to *CMS-mini*.

Currently, when deployed and started for the first time, *CMS-mini* will fetch general information about the course and at least one instructor of it. This information should be prepared by the instructor and shared with *CMS-mini* via Google Drive. Upon successfully parsing the xml data, the application will be ready to accept the instructor to login. Figure 4 shows the user interface of the development version of *CMS-mini* (*http://younes-mini.appspot.com/home.jsp*).

In this screen capture it is possible to note that the application parsed the basic information of the course (course title *"Computer-based Simulation"* and code *"401481"*) and placed them in the corresponding fields in the file "`home.jsp`". From this point on, the instructor will be able to login and import the detailed information of the course and the attendees. This information will be saved in xml format that is referred to inside the `<extension>` tags in the xml file parsed in the first place.

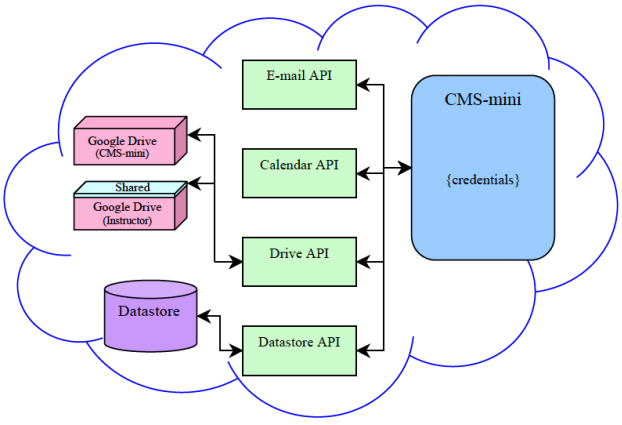

Figure 3: CMS-mini structure

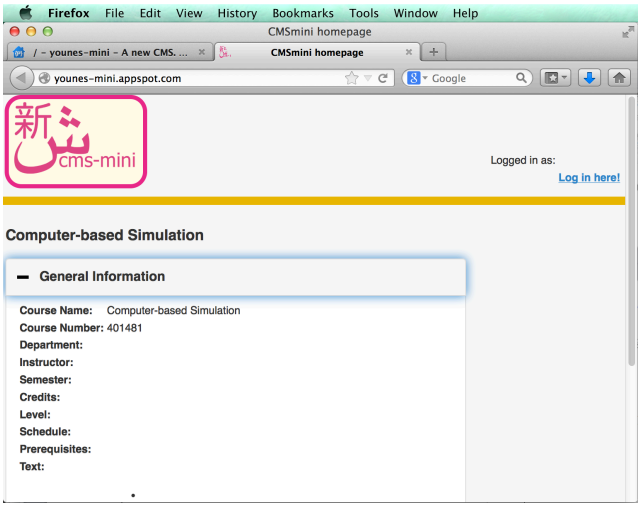

Figure 4: Development version of *CMS-mini*

5. Concluding Remarks

This submission introduces the situation that led to the need for a way for distributing course materials during a catastrophic situation that led to abandoning the university campus and being unable to use the e-learning system of the university. It shows how the teaching staff initially managed to do this in person in order to proceed the semester and the difficulties they faced.

Then we introduce the new approach to delivering the contents via central storage on Google Drive and show the feedback of this experience. Students faced a problem of being unable to upload their reports and assignments to course folder shared with them, and instructors did not have the possibility to track the course activities.

Finally, we introduce the latest approach that is based on "Cloud Computing" environment offered by Google "Google Apps Engine". This approach introduces the design of a new web-based application "*CMS-mini*" for managing course contents. The design is still in the development phase, but progress is being made every day.

Reference

1) Jadeja, Y. and Modi, K.: Cloud computing - concepts, architecture and challenges, 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET 2012)
2) Wikipedia: Cloud Computing Layers, http://upload.wikimedia.org/wikipedia/commons/7/7e/Cloud_computing_layers.svg
3) Wikipedia: Google Apps, http://en.wikipedia.org/wiki/Google_Apps
4) Google Inc.: What Is Google App Engine? https://developers.google.com/appengine/docs/whatisgoogleappengine
5) Microsoft: What is Azure, http://azure.microsoft.com/en-us/overview/what-is-azure/
6) Amazon Web Services: Amazon EC2, http://aws.amazon.com/ec2/
7) IMS Global Learning Consortium, Inc.: IMS Enterprise Specification, http://www.imsglobal.org/enterprise/

Acknowledgments This submission is based on a research supported by the Japanese Society for Promotion of Science (JSPS). The research was carried out with the support of a Grant-in-Aid for Scientific Research.