CREATING AND IMPLEMENTING CLOUD-BASED SIMULATIONS TO ENHANCE A MULTIVARIABLE CALCULUS COURSE

Joe Guarino[1], Seung Youn Chyung[2], Charles Adams[1], Rey DeLeon[1], Marion Scheepers[3], Francisco Castellon, Michael G. Wiedenfeld[4], Paul Williams
[1] Mechanical & Biomedical Engineering, [2] Instructional & Performance Technology,
[3] Mathematics Department, Boise State University
[4] Kuna High School

Abstract

A Cloud resource at Boise State University was used to enhance a large section of Multivariable and Vector Calculus (MATH 275) taken by engineering majors. This section was developed to deal with rapidly-increasing engineering enrollment. Our hypothesis was that curricula could be successfully delivered to a very large class (120 students) by augmenting instruction with Cloud resources and Clicker technology. Interactive exercises, hosted on the Cloud, were assigned instead of traditional text-based homework. Exercises were developed by a team of faculty and graduate students funded by a Hewlett-Packard Labs grant[1]. Exercises were created using MATLAB[2] and Working Model[3] software. Student satisfaction and perception of learning were measured using Clicker-based surveys associated with each exercise.

Cloud computing resources hosted on university workstations provide access to licensed software used by STEM students. University students access our cloud resource using the same user id's and passwords that they use to access other University resources. Remote Graphics Software (RGS)[4], available from Hewlett-Packard, enables students to remotely access any software made available to them on the cloud. Moreover, RGS enables the students to work together on the same file. Finally, RGS Sender software enables the host computer to do most of the video processing, so that the remote user can run graphics-intensive software using a low-end PC or thin client without performance impairment. The cloud was hosted on 16 Blade workstations provided by a Hewlett-Packard Innovations in Engineering (IOE) award[5].

Providing easy access to our cloud resource was fundamental in achieving our goals. We developed a rubric for accessing and using our cloud resource. Clicker surveys conducted during classes at the beginning, middle, and end of the semester provided data elucidating student opinions on accessing and using the cloud resource. Comments were also elicited during the end of semester course evaluation. Results showed students became more comfortable with the cloud resource as the semester progressed. Almost all of the students were comfortable with accessing and using the cloud resource by the end of the semester. Our template for providing cloud resources might be useful for others considering the implementation of cloud technology.

Introduction

The technology and resources provided by cloud computing may eventually transform the way we teach and learn. This new resource is now in the implementation stage in higher education. Even a cursory literature search results in a dizzying array of lexicon which can be very confusing. A cloud computing resource has different meanings to end users, developers, and information technology administrators. One convenient definition is provided by Mell and Grance (2009)[6] and quoted below:

"a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be

rapidly provisioned and released with minimal management effort or service provider interaction."

Cloud computing architecture is conveniently described using a layered model presented by Zhang (2010)[7] and summarized below, from the highest layer (application) to the lowest layer (hardware):

1. Application (programs accessed by users)
2. Platforms (software framework)
3. Infrastructure (computation and storage)
4. Hardware (CPU, memory, etc)

Our project uses an *application cloud*, which provides software-as-a-service (SaaS). Our application cloud provides access to software applications and also facilitates shared use of the applications, allowing students to collaborate remotely on the same assignments. Unlike a traditional server, which provides multiple users access to multiple instances of a program, our cloud resource provides multiple users access to a single instance of a program. This type of cloud can be easily established at other universities by installing RGS software on any workstation with a Windows or UNIX operating system.

The most important aspects of cloud resources pertaining to our project are remote access to university software at any time, and collaboration of our students in a virtual environment while working on the same file. The general requirements for students accessing and using our cloud resource are listed below:

1. Installation of facilitating software
2. Establish user names and passwords
3. Verify available cloud hosts
4. Share and Disconnect the cloud host

It's also very important to establish protocol for use of the cloud resource, and also to provide detailed instructions for accessing and using the cloud resource. Finally, we implemented a formative survey plan to assess our effectiveness in delivering the cloud resource.

Methods

The methods used to implement each requirement are detailed below.

<u>Installation of facilitating software</u>

Front-end processing and small commitment for client resources are technical hallmarks for an effective cloud resource. Remote Graphics Software (RGS), which can be purchased from Hewlett-Packard, was used to create our cloud resource. At the time of this article, RGS is available for Windows and UNIX systems. RGS Sender was installed on each of 16 Blade workstations. Computers hosting a cloud resource need no keyboards or monitors; therefore, they are often conveniently housed in a rack configuration. RGS Receiver software must be installed on client computers to work with RGS Sender. RGS Receiver is currently available for Windows and UNIX systems at no charge from Hewlett-Packard. RGS software creates an excellent cloud resource because almost all of the processing, including video processing, is done by the host computer. RGS Receiver has a very small "footprint" on the client machine. Complete instructions for downloading and installing RGS Receiver were provided on the MATH 275 website, and an instructive video was provided on the Engineering Learning Community for Idaho (ELCI) website[8]. Figure 1 shows the RGS Receiver window used to connect to a host computer.

<u>Establish User Names and Passwords</u>

User names and passwords are needed to access the cloud resource. We found it most convenient to use the standard user names and passwords that the students use to access university computing resources. We note that it is possible to limit Internet access through the cloud hosts by using designated user names and passwords.

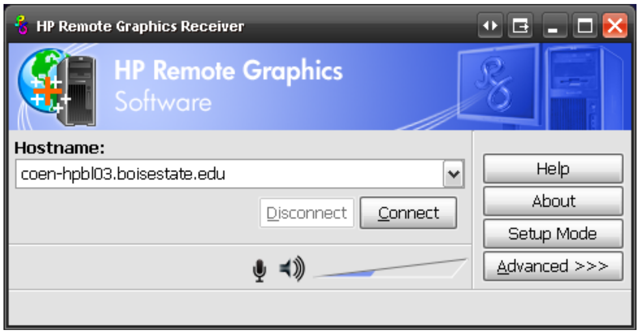

Figure 1: RGS connection window.

Verify Available Cloud Hosts

Session management software can be used to direct users to available host computers. Another approach is to use a program such as LABSTATS[9]. LABSTATS provides a screen showing the state of use of each cloud host, as shown in Figure 2. In our example, red means unavailable, blue means in use, and grey is available.

Share and disconnect the cloud host

A student logging onto an open cloud resource will see a desktop image from the cloud host. The host desktop appears as the blue inset screen in Figure 3 shown below. The student can run any program on the host; in Figure 3, Working Model and MD SOLIDS[10] are both open on the cloud host. The first user to log on to a cloud host is the primary user. The primary user can allow access to secondary users, who can then observe the same desktop. The primary user can also enable mouse sharing, which allows other users to control the mouse. Therefore, multiple remote users can interact with open programs on the same cloud host. We recommend that students open chat windows on their client computers to communicate with other users logged in to the cloud host.

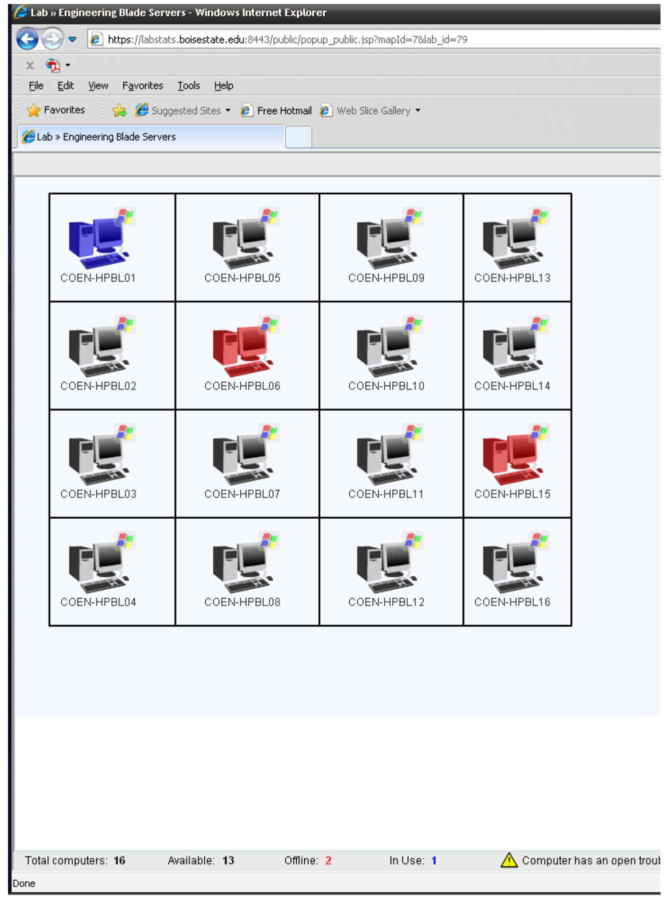

Figure 2: LABSTATS Screen showing host availability.

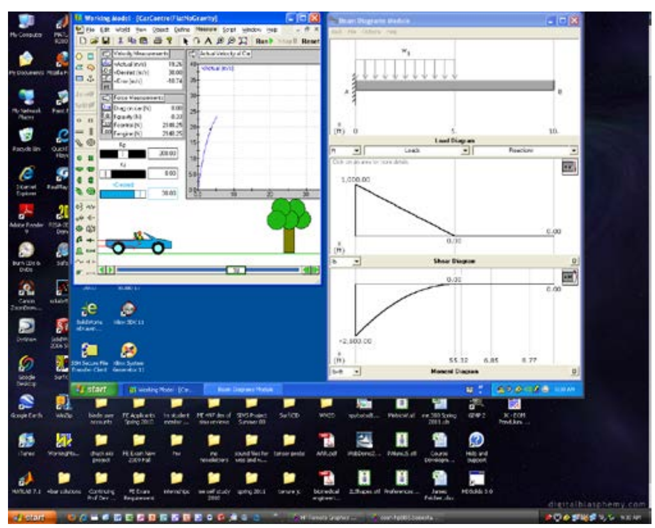

Figure 3: Cloud Host desktop insert on client desktop (dark background).

The primary user, or whoever is controlling the mouse, can log out of a session, which closes the session for all users. There is an adjustable time-out facility which allows the primary user to log back on after being disconnected for a pre-set time period. This feature retains program states in the event of interrupted internet access. Unfortunately, if the primary user disconnects instead of logging out of a session, the host computer cannot be accessed by someone else during the pre-set time period. Therefore, it must be stressed to users that disconnecting is not the same as logging out.

Simulations for the cloud

There are many interactive programs that facilitate teamwork in a cloud resource. Interactive simulations for MATH 275 were developed using Working Model and MATLAB. MATLAB also supports programs that download student performance metrics to a spreadsheet. The routine developed by our team is included in Appendix 1.

Results

Three formative clicker surveys conducted during the course included three questions related to accessing and using the cloud resource. The first, second, and third clicker surveys were administered at the beginning, middle, and end of the semester. The three survey questions pertaining to accessing and using the cloud resource are shown below:

1. How easy or difficult was it to connect to the Blade server?

2. Once you log into the Blade server, how easy or difficult was it to start a simulation program?

3. How easy or difficult was it to connect remotely with classmates?

Students responded to the survey questions using the following Likert scale:

1	2	3	4	5
Very Difficult	Difficult	Easy	Very Easy	Not applicable (I have not connected / run a program/ connected with classmates

Results from clicker surveys are shown in Figures 4, 5, and 6.

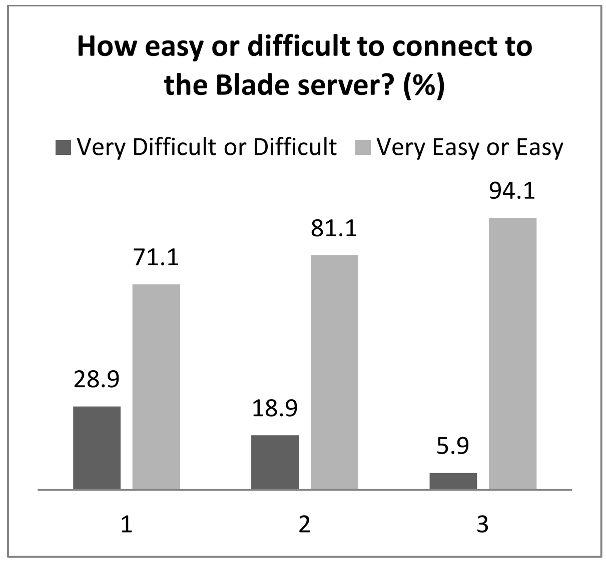

Figure 4: Connecting to the Server.

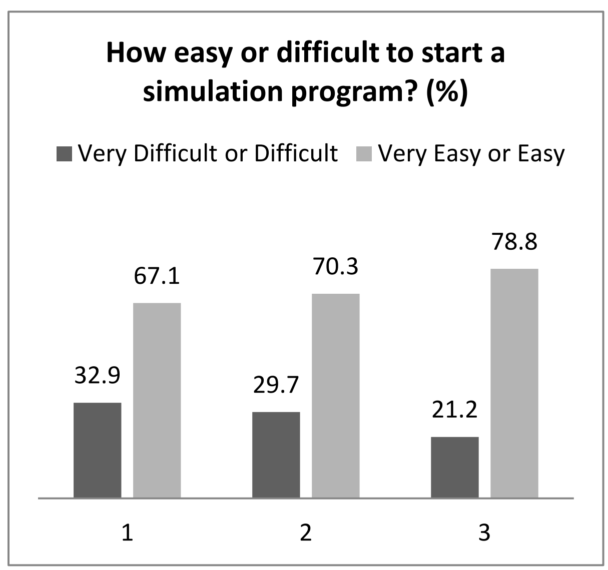

Figure 5: Starting a Simulation.

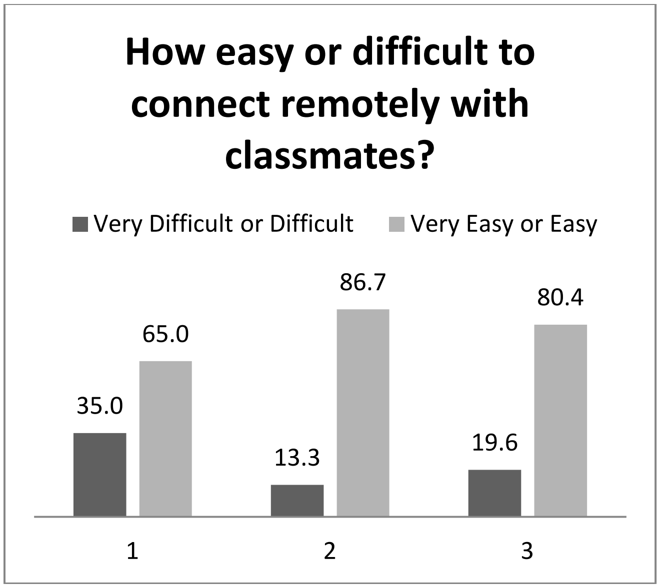

Figure 6: Connecting Remotely with Classmates.

Responses to the first survey question show that student ability to connect to the cloud resource increased through the semester. At the end of the semester, over 94% of the students found it easy or very easy to connect to the cloud. Students indicated that starting a simulation program was more challenging than connecting to the cloud. Their ability to start a simulation program increased throughout the semester as well; however, we are concerned that more than 20% of the students found it difficult or very difficult to start a simulation program at the end of the semester. We feel that connecting with other students during the simulations by using chat windows promotes an engaging learning environment critical to effective use of our cloud resource. We are encouraged that over 86% of the students found it easy or very easy to connect remotely with classmates by the middle of the semester; however, we are concerned at the paradoxical drop in this metric to 80% by the end of the semester.

Discussion

We consider the first deployment of our cloud resource a success. We learned the following lessons from the first phase of our project which might be helpful to others:

Considerable overhead in time and expertise must be invested at the beginning of the course.

Downloading and installing the software used to access the cloud resource was by far the largest source of frustration to our students. While over 70% of our students had little or no trouble at the beginning of class, the rest of the students experienced significant difficulty. We found that all of the issues were due to lack of computer skills. These issues were easily resolved; however, a considerable devotion of time by the investigators was required. We also provided an informative video with instructions on downloading and installing RGS and accessing our cloud resources. We also linked a LABSTATS page showing host availability (as shown in Figure 2). Both the informative video and the LABSTATS page appear on our Engineering Learning Community for Idaho (ELCI) website.

Computer Resources must be provided to all students.

A small number of students had limited or no access to their own computers. Further, RGS Receiver was not compatible with MAC operating systems at the time of this study. Therefore, computer access to the cloud resource had to be provided. Our Engineering Learning Community for Idaho (ELCI) lab, provided by a Hewlett-Packard Innovations in Engineering award, was made available for students to access our cloud resource.

Students must be individually accountable for completing cloud-based assignments

Our team developed MATLAB routines to track completion of weekly assignments. Each student working on a simulation was required to enter his or her ID number. This method can clearly be circumvented, but it demonstrates to the student the importance of individual accountability.

Future Work

MATH 275 is a prerequisite for several engineering courses, including Fluid Mechanics and Electromagnetic Theory. We will survey engineering faculty to determine if student performance in these courses was affected by our new approach. The refinement of a translator application for Google Wave (Rosie) provides the exciting prospect of international and multicultural collaboration of STEM students at all levels.

Acknowledgements

We wish to thank the Hewlett-Packard Innovations in Engineering (IOE) program and the Hewlett-Packard Research Labs program for their generous and invaluable support.

Bibliography

1. "Rising Cloud Project," Hewlett-Packard Labs Innovation Research Program, May, 2010

2. MATLAB is available from MathWorks, Inc., 3 Apple Hill Drive, Natick, MA, 01760-2098, USA

3. Working Model is available from Design Simulation Technologies, Inc., 43311 Joy Road, #237, Canton, MI, 48187, USA

4. Remote Graphics Software (RGS) is available from Hewlett-Packard Company, 3000 Hanover Street, Palo Alto, CA, 94304-1185, USA

5. "Engineering Learning Community for Idaho," Hewlett-Packard Innovations in Education (IOE) Grant, May, 2009

6. Mell, Peter and T. Grance, "Draft NIST Working Definition of Cloud Computing," from http://csrc.nist.gov/groupps/SNS/cloud-computing/cloud-def-v15.doc , August, 2009

7. Zhang, Qi, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," Journal of Internet Server Applications, Springer, 1:7-18; 2010

8. "Engineering Learning Community for Idaho Website," http://coen.boisestate.edu/elci

9. LabStats is available from Computer Lab Solutions, 255 B St, Suite 207, Idaho falls, ID 83402

10. MDSolids is available from Timothy A. Philpot: http://www.mdsolids.com/

Biographical Information

Joe Guarino is a Professor of Mechanical and Biomedical Engineering at Boise State University. His research interests include educational aspects of cloud computing, vibrations, acoustics, and dynamics.

Seung Youn (Yonnie) Chyung is a professor in the Department of Instructional and Performance Technology in the College of Engineering at Boise State University. She teaches graduate-level courses on evaluation methodology and e-learning. Her research interests include the development of self-regulated e-learning strategies for adult learners and the pedagogical use of technology.

Charles Adams is an Undergraduate Research Assistant, Mechanical & Biomedical Engineering, Boise State University Boise, ID.

Anthony Rey DeLeon is graduate research assistant with the Mechanical & Biomedical Engineering Department at Boise State University. His current research involves GPU-accelerated computational fluid dynamics. Past research included the software development of MATLAB simulations for abstract math concepts deployed on cloud computing resources.

Marion Scheepers, Department of Mathematics, Boise State University. Educational Background: Ph.D. in Mathematics (1988) from The University of Kansas. Advisor: Fred Galvin. Current Employment: Professor, Department of Mathematics, Boise State University, Boise, ID 83725

Francisco Castellon, Systems Administrator, College of Engineering, Boise State University.

Michael Wiedenfeld has taught science at Kuna High School in Kuna Idaho for 20+ years. Michael currently teaches Chemistry and Introductory Research/Pre-Engineering. Michael earned his Masters Degree in

Educational Technology from Boise State University in 2002 and his undergraduate degree in secondary education (Emphasis-Physics) from the University of Idaho in 1989. Michael has served on the Idaho Science Teachers Association board for 9 years including as president in 2005. Michael currently serves as the Kuna School District Science Curriculum Chair and the Kuna High School STEM academy coordinator. Michael has taught numerous teacher training workshops on the integration of technology into the classroom to k-12 teachers.

Paul Williams, Graduate Research Assistant, Stem Station, Boise State University.

Appendix 1

Grade recording routine for MATLAB™ :

```
%% Created by Charles Adams %%
%% Mechanical & Biomedical Engineering %%
%% Boise State University %%
%% GradeIDSingle Function %%
% The purpose of this function is to access a Microsoft
Excel sheet and put
% marks on the sheet that signify grading and completion
of program
% exercises for a single user. This currently can be
completed from a
% campus computer or Blade server only. %

function GradeIDSingle(UID,start)
%% Null-ID Check %%
if UID == 111
   return
else
%% Grading Variables %%
% Looping Variable %
c = 0;
% Path %
path = '\\coen-
maple\common\Everyone\CAdams\Programs\ExcelTest\S
coresheet.xls';
% Password for document %
passcode = 'passcode';
% Select correct tab %
tab = 'Scoresheet';
% Select correct range of values %
datarange2 = 'B2:J51';
%% Excel Object Initialization %%
% Excel Startup %
e = actxserver ('excel.application');
eW = e.Workbooks;
% Opens document in background %
while c == 0;
   % Try-Catch loop to help deal with possible collisions
%
try
eScore = eW.Open(path,[],false,[],passcode);
break
catch
   msgopenerror = msgbox('MATLAB is having trouble
opening the grading grid, please press ok to try again, if
error persists, contact instructor.','Collision Error');
   waitfor(msgopenerror);
   c = 0;
end
end
%% Matrix Collection %%
% Selected Tab %
Current = eScore.Sheets.Item(tab);
% Selected Range %
RangeObj = Current.Range(datarange2);
% Gathers Range Data %
ExMat = RangeObj.Value;
% Converts into MATLAB matrix form %
MMat = cell2mat(ExMat);
% Finds the row the User is associated with %
[row] = find(MMat == UID);
%% Grade & Timestamp Student ID %%
% Gathers completion value %
comp = MMat(row(1),2);
% Clocks Time Stamp %
endtime = fix(clock);
% Checks completion value %
if comp == 0
   % Checks Completion Box %
   MMat(row(1),2) = 1;
   % Time Stamp of Completion %
   MMat(row(1),3:8) = endtime;
end
% Stamps time ellasped on program in minutes %
MMat(row(1),9) = MMat(row(1),9) +
etime(endtime,start)/60;
%% Excel Export %%
% Convert back to cell matrix %
MMat = num2cell(MMat);
% Inserts new values in range %
Current.Range(datarange2).Value = MMat;
% Saves and overwrites the Excel document %
eW.Item('Scoresheet.xls').Save;
% Closes the actual Excel document %
eW.Item('Scoresheet.xls').Close;
% Quits Microsoft Excel %
e.Quit;
% Deletes the associated object %
e.delete;
% Clears the object from MATLAB %
clear e;
end
%% End Function %%
```