Databases in the Cloud: A Status Report

Edward P. Holden  Jai W. Kang  Geoffrey R. Anderson  Dianne P. Bills
1.585.475.5361  1.585.475.5362  gra4497@rit.edu  1.585.475.6791
edward.holden@rit.edu  jai.kang@rit.edu  dianne.bills@rit.edu
B. Thomas Golisano College of Computing & Information Sciences
Rochester Institute of Technology
Rochester, New York 14623

ABSTRACT
This paper updates an earlier paper on the use of cloud computing in database curriculum. That paper described a curricular initiative in cloud computing initially intended to keep our information technology (IT) curriculum at the forefront of technology and to give students the flexibility to work at any location, not just our labs. Currently, our IT degrees offer extensive database concentrations at both the undergraduate (BS-IT) and graduate (MS-IT) levels. This paper reports on the results of two years of operation using a cloud provider for lab exercises in our Database Architecture and Implementation course. It discusses the benefits gained and concludes with an overview of a new cloud deployment strategy that improves disaster planning for our curricular infrastructure and provides an extension to another campus.

We discuss how the Cloud Vendor Selection Model, proposed in our previous paper, shows the ways in which the different layers of cloud services interact with each other. Plus we show how the different categories of cloud users in this model can be supported by different educational tools to meet course objectives.

Finally, we discuss the various issues and challenges that we have experienced when implementing cloud solutions in an educational environment.

Categories and Subject Descriptors
K.3.2 [Computer and Information Science Education]: Curriculum
General Terms
Management, Performance, Economics, Reliability.
Keywords

1. INTRODUCTION
In the fall quarter of 2009, we began using the Amazon Web Services (AWS) cloud to support laboratory exercises in our database administration course, Fundamentals of Database Architecture and Implementation (4002-485). The primary driver for this change was to expose students to new technologies, in this case a cloud infrastructure. This foundation, together with the opportunity to use other types of cloud services like Google Docs and MapReduce in other courses, gives students a perspective on the variety of services that are available in the cloud. The secondary benefit was flexibility: both for our curriculum and for students to do their work from other locations, not just in our computer labs. Our internal database lab setup gives students root access to the machines; so, we needed to restrict outside access to these labs.

The original design, described in our earlier paper [2], has worked well since its inception. However, changes to the Amazon Machine Image (AMI) and to our database image-generation process – to expand capabilities and to solve small problems, such as providing additional scripts for students or decreasing the amount of time needed to start up a database instance – have been implemented each term. This paper also discusses an example implementation of the cloud-based curriculum design model proposed in our earlier paper and provides a status report on our experience with this curricular design to date.

2. THE FIRST-COURSE EXPERIENCE
Although the technologies used in the cloud are not necessarily new, there are many documented advantages to using cloud technology, for example, those listed in Cloud Computing – The Business Perspective [5] in which the authors identify the following key points:

1. It lowers the cost of entry for smaller organizations with computer intensive applications.
2. It can provide almost immediate access to hardware and software resources without upfront capital investment.
3. It can lower barriers to innovation.
4. It makes it easier for an organization to scale their services to changing needs.
5. It makes it possible for new classes of applications to be developed and to deliver services that were not previously possible.

We have found the above points to be valid for our educational cloud implementation. On the whole, AWS has been very reliable. However, cloud-based curriculum has not allowed us to reduce lab costs, since we have maintained our current lab setup so that students can do exercises in class with the instructor. Plus, maintaining our original lab-imaging process has given us a reliable backup strategy in case of an AWS failure; this is further discussed later in this paper.

The cost of entry has been low (point 1). Our students have typically spent from $8 to $12 each to use the AWS environment for class work, for a total of $240 to $300 for each class taught. Amazon [1] has continued to provide grant funding for students to work in their cloud environment through their AWS in Education grants (http://aws.amazon.com/education/). These grants provide $100 of computing time and other cloud resources for each student. As an aside, faculty and students can also apply for larger research grants.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGITE’11, October 20-22, 2011, West Point, New York, USA.
Copyright 2011 ACM 978-1-60558-765-3/09/10...$10.00.
We will be saving money with additional deployments of this course (points 1, 2 and 4), however. In the fall of 2011, this course will be offered at our American College of Management and Technology in Dubrovnik, Croatia. Without AWS, this would have required the duplication of our entire database lab environment in Dubrovnik, with significant capital cost and development expense. Our cloud solution is to create an AWS machine image in the AWS EU Region (Ireland).

Although most students only spent about 10% of their $100 AWS grants for coursework, one positive consequence has been student exploration and innovation (points 3 and 5). Some students have used their grant funding to explore other uses for the AWS cloud technology and its available software. Spending up to $60 of their grants on average, these students have developed Web sites or explored other technologies of their interest, such as clustering or MapReduce.

One other change is that, originally, we only supported Microsoft Windows clients in our database labs. These clients require downloading PuTTY for the secure shell (SSH) connection to the AWS server and Xming to handle X11 forwarding. We have now extended support (points 4 and 5) to Mac and Linux clients, since more students are using these systems. Most Macintosh and Linux clients natively support SSH and X11 forwarding; so this only involved a change in our lab instructions.

3. USING THE CLOUD-VENDOR SELECTION MODEL

Based upon the three-dimensional Cloud-Vendor Selection model (CVSM) previously proposed [2], we selected an IaaS (Infrastructure-as-a-Service) cloud provider, AWS, to support laboratory work in the Database Architecture and Implementation course. CVSM defines a three-dimensional selection matrix (cloud service vendors; industry types; and cloud-user types) that can be used to determine the most appropriate service provider for a specific type of user in a particular industry [2, Figure 1]. We now discuss how CVSM can be applied to curriculum design in higher education.

The Cloud Security Alliance [3] presents the relationships and dependencies between cloud service providers in its Cloud Reference Model. As described, IaaS, the foundation of all cloud services, is a low-level hardware virtual machine for the delivery of servers, storage, network infrastructure and operating systems. PaaS (Platform-as-a-Service) is built upon IaaS to offer developers the ability to create applications with programming languages and database support. SaaS (Software-as-a-Service) is built upon PaaS, and thus IaaS, to provide a high-level framework for the applications delivered to the end users of a given service. The key feature being that cloud service architecture, as shown here in Figure 1, is a tri-level superset of functionality with IaaS as the innermost, or foundation, technology subset; and educational curriculum may be at or cross multiple levels.

Combining this perspective with our model, we then applied CVSM to various course areas in our industry (Education, college information technology curriculum) such as Web, database, software development, networking and security, etc. After surveying the objectives of various courses across these topic areas, we found that objectives tend to overlap. Therefore, a condensed list of unique objectives was sufficient to map to the CVSM Service Providers and Users dimensions; and the results are discussed here for our database curriculum.

Our database courses have three types of Users: Administrators, Instructors and Students. The specific course objectives for our curriculum are summarized in Table 1, with the following domain business rules:

1. An Administrator sets up an AMI for a given course.
2. An Administrator sets up the accounts for Students, who log in and perform laboratory assignments and practicums.
3. An Instructor checks the correctness of the image for laboratory assignments and practicums, and ensures that the machines are ready for Students to use. An Instructor may make changes to the image, as required.

So, for CVSM in this case, Administrator is a Service Author (the user who develops an image to meet either the needs of a specific end-user or a logical subset of composite solutions), Instructor can be a Service Author or an End User, and Student is the designated End User. For our database curriculum, an Administrator customizes an IaaS image for Instructors and Students to use. Once the image is ready to use, it is a PaaS. Occasionally, students make changes to the image, as required. So, for CVSM in this case, Administrator is a Service Author (the user who develops an image to meet either the needs of a specific end-user or a logical subset of composite solutions), Instructor can be a Service Author or an End User, and Student is the designated End User. For our database curriculum, an Administrator customizes an IaaS image for Instructors and Students to use. Once the image is ready to use, it is a PaaS. Occasionally, students make changes to the image, as required. So, for CVSM in this case, Administrator is a Service Author (the user who develops an image to meet either the needs of a specific end-user or a logical subset of composite solutions), Instructor can be a Service Author or an End User, and Student is the designated End User. For our database curriculum, an Administrator customizes an IaaS image for Instructors and Students to use. Once the image is ready to use, it is a PaaS. Occasionally, students make changes to the image, as required.

Table 1 provides specific examples of how our database curriculum Users map in the CVSM model [2]. As can be seen in Table 1, we have mapped our database curriculum domain users to the CVSM Users categories for our database course objectives. In the next section we elaborate on the role of our Administrator user in the creation of a database PaaS image for the Fundamentals of Database Architecture and Implementation course and share some of our experiences.

4. AN IMPROVED CLOUD CURRICULUM ARCHITECTURE

As mentioned in the introduction to this paper, basing our database curriculum on an Amazon Machine Image (AMI) has
worked well since our first implementation in 2009 (general process available at [14]). However, our modifications to the base AMI used and our course image-generation process have changed over time in response to various issues and needs.

Although very reliable, there have been two outages that have impacted the AWS cloud in the US East Region (Virginia) where our image is stored. One outage was the well-publicized, multi-day outage caused by the failure of the AWS EBS (elastic block storage) subsystem [4]. This had little impact on our AWS environment since most students were able to access our AMI image with little difficulty. The other outage occurred during a practical exam; and we were not adequately prepared for it. For our lab backup strategy, we have our own Linux-based Oracle servers in a VMware image on our lab Windows machines. This image is accessible to our students for use during in-class activities. So during the outage, students were able to switch to this image for the practicum. There were unanticipated issues, however. Unfortunately, a couple of changes to our AWS environment had not been implemented in the lab VMware image. This resulted in a loss of about fifteen minutes during the exam to explain the differences plus additional stress for students. This experience, and the need to support our database curriculum at multiple locations, meant that we needed a new integrated image-creation and backup strategy.

To solve the backup problem, provide image consistency, and support the migration to Croatia, we have developed a new process that generates both the US East and EU images from a local VMware image. The benefit of this approach is that we will have images available at the locations needed and additional backup environments that can be used by students in both locations. Plus, we will only need to build one image and then clone it for use in multiple environments. As part of this effort, we conducted research into the additional technologies provided by AWS and learned that the AWS API has been updated to allow for a new type of AMI root-device storage: Elastic Block Store (EBS) [6, 12]. The advantages and disadvantages for us when deploying AMIs under this new storage mechanism are detailed in Table 2 below.
<table>
<thead>
<tr>
<th></th>
<th>EBS-backed AMI</th>
<th>S3 Instance-store AMI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Boot Time</strong></td>
<td>1-2 minutes</td>
<td>10-15 minutes</td>
</tr>
<tr>
<td><strong>Creation/Deployment Time</strong></td>
<td>Less than 60 minutes</td>
<td>1-4 hours</td>
</tr>
<tr>
<td><strong>Creation/Deployment Dependencies</strong></td>
<td>Existing EBS-backed AMI</td>
<td>RAW image file with appropriate modules or existing S3 AMI</td>
</tr>
</tbody>
</table>
| **Creation Process**          | ● Register AMI with existing EBS snapshot (Note: you can create EBS Snapshots from existing EBS volumes.) | ● Bundle RAW image  
● Upload bundle to S3  
● Register AMI from uploaded bundle stored in S3 bucket |
| **Stoppable**                 | Yes             | No (can only be terminated)   |
| **Deployment Across Different Regions (US East/West, EU, & Asia)** | Need to create and maintain AMI separately for each region | S3 instance-store AMI are accessible from all regions. |
| **Size Limit**                | 1 TB            | 10 GB                         |
| **Charges**                   | ● Standard EBS volume charges  
● Hourly instance charges  
● Restart/Stop charges     | ● Standard S3 Storage charges  
● Hourly instance charges  |

Based on the information in Table 2, we opted to migrate to an EBS-backed AMI and selected an existing AMI that was preloaded with Oracle Enterprise Linux. We then modified the image to meet our criteria for a virtual environment within which our students could operate. From this we created a new EBS-backed AMI so that we could easily manage our custom AMI and open it for student use in the Fundamentals of Database Architecture and Implementation course, each lab is dependent upon the previous lab. Under the S3 instance-store AMI environment, students needed to shut down their database, backup the database onto S3 volumes, and terminate their instance between sessions. The problem with this approach was that all references to the student’s AWS instance disappeared which would add additional startup time for students when they returned to work on their labs. This is because the student would need to start up a fresh AWS server instance, restore their database from the last backup on S3, and then start up an instance of the Oracle database. This process would generally take up to fifteen minutes.

Using an EBS-backed AMI streamlined this process. Now when finished with a lab, students still shut down the Oracle database instance and backup their database to S3, but instead of terminating the AWS instance, they simply stop it. This leaves the instance intact with its data. To resume work, the student restarts the already existing AWS instance and then starts up the Oracle instance. A restore of the database is not needed, which saves time, although doing a backup is still available if needed.

It was under this revised AMI architecture that we experienced the AWS latency issues and failures mentioned above. This emphasized the necessity of having a fallback solution in the event of any availability problems with the AWS cloud computing infrastructure. The solution we decided upon was to maintain a copy of our original VMware image along with the EBS-backed AMI so that, in the event of an AWS EC2 failure, our students would have the option of performing their work in lab with an environment similar to that of AWS’s EC2. The consequence of this approach is that we have to manage more than one virtualized environment image and maintain image consistency across multiple environments.

To streamline this process and to eliminate image inconsistencies, our database graduate assistant constructed two Bash shell scripts to support the process of managing our virtual machine images both in the labs and in the cloud: first, a script that converts a VMware disk (VMDK) image file to the appropriate format to be deployed on AWS EC2 [17]; and then a second script that allows us to easily deploy the converted image file to multiple AWS regions [16]. The construction of these scripts was necessary, in part, because the process of importing a custom VMware virtual machine image into AWS EC2 environment was limited to Windows Server 2008 at the time of this writing [9].

The second script (inspired by [8], [13]) creates an EBS-backed AMI. It implements a pre-existing Amazon process in which a loopback file containing a Linux operating system is bundled, uploaded to an S3 bucket, and is then registered as an instance-store AMI [10]. The script then launches the instance-store AMI, verifies that it is running, uploads and runs the “instance-to-ebami.sh” conversion script from [11], and finally creates an EBS-backed AMI using the volume that was populated by [11]'s script. Upon completion or error, the script will terminate any running instances it launched and display information detailing its completion status.
As a result, our database AWS AMIs will be deployed from a common image, ensuring consistency among the images as illustrated in Figure 2. Our Administrator will make all changes to the local VMware image which is then used to create and deploy AMIs to any desired AWS region (US East and EU in our case). Our students on the RIT campus will primarily use the AMI in the US East region with the EU region image or a local VMware image as a fallback environment. Our students in Croatia will primarily use the AMI in the EU region with the US East region as their fallback environment.

**Figure 2: Deployment and Usage of Machine Images**

5. **ISSUES AND CHALLENGES**

So far, our use of cloud computing has been a success for the lab exercises in the Fundamentals of Database Architecture and Implementation course. There are, however, a couple of issues that we still need to address.

First, we need to remember that students are students and will make mistakes. Although the vast majority of students have had no problems, there have been a couple of challenges to using the cloud for some students. These can be categorized as follows:

- Billing begins when a student starts an AMI instance. Some students have forgotten to stop their instances when they complete a lab; so their billing continued. In most cases, there was no actual bill for the student due to the $100 Amazon grant, but we did have a case this year in which a student exceeded her $100 grant. It would be beneficial to have a way to remind students that their instances are still running. Our original design was to send an email from the running instance to students every hour, but our email system identifies these emails as spam and does not allow them through.

- At the end of each term, students should delete their data storage on S3 or they will be billed for this disk usage. Some students have received charges when their grants were exceeded or expired. We have added instructions to the course that explains what needs to be done, but some students have not followed them. AWS has been very good about refunding some or all of the charges to these students.

- One student used an instance that was too large. Our lab exercises specify that a small instance should be used which is currently billed at a rate of $0.085 per hour. There are presently other Amazon image sizes: large, billed at $0.34 per hour; and extra large, billed at $0.68 per hour. Naturally, the more expensive sizes use up the grant more quickly and thus students could accrue charges that exceed the funds provided by their grants.

The other areas that remain a challenge are backup and recovery, and timely disaster management. As mentioned earlier, our disaster recovery images were not kept consistent. This is rectified as described in Section 4, but vigilance – along with increased automation – is key to avoiding this problem in future. Students are responsible to back up their data after each lab, and this is specified in the lab instructions. However, students can forget or skip the process when busy. So, to help mitigate loss of student work, we have a backup image that they can restore should their data be lost. It is incumbent upon us, however, to continue to impress upon them that backup and disaster planning are just as critical in education as they are for business and industry.

A final concern with our current process is the need to plan ahead for the administration time needed by our Administrator, as well as for that for an Instructor, to support the image creation and management process just as we would with any local image implementations that supports curriculum. This cloud image implementation process is non-trivial.
6. FUTURE DIRECTIONS
Our current focus has been on improving the use of cloud computing in one course only. As shown in Table 1, this technology could be applied to supporting other courses as well. Course objectives can be mapped to the User categories in our Cloud Vendor Selection Model which then determines the appropriate types of cloud providers. Our next work should focus on extending the use of the cloud to other curriculum, particularly for programming courses, other database courses, and Web courses. We may also consider using cloud technology to extend this course, and others, to students in off-campus locations. Plus, there is also an opportunity to provide supportive learning environments for service courses targeted at non-technical students.

7. ACKNOWLEDGEMENTS
We wish to express our thanks to Gary Scarborough, our database system administrator, for his ongoing efforts on behalf of advancing our database curriculum. We would also like to thank Amazon.com for their continuing support of our curriculum in the cloud.

8. CONCLUSION
Our implementation of cloud computing, in one database course so far, has been very successful. Students have been excited about the opportunity to learn about cloud technologies and about the flexibility to work on their lab exercises outside of our lab environment. Some students have been inspired to try out new cloud-based implementations beyond their course requirements. Our students also benefit by being able to list their cloud-computing experience on their resumes. Database students – both undergraduate and graduate – can cite their understanding of cloud technologies and their experience with AWS. Some graduate students use the technology in their capstone work. Additionally, the database graduate assistants who have been actively involved in the design, configuration, and set up of AMI server instances and with the integration of this technology into our curriculum have gained extremely valuable experience. These graduate assistants are also actively involved in managing our labs and with tutoring students; as a result, their excitement and willingness to share their cloud skills has definitely enriched our database learning environment. This has clearly made all of our database students more attractive to potential employers.

Another important outcome from our experience is the realization of the critical need for solid backup, recovery and disaster recovery processes. Students can now backup their image but still maintain the state of their servers between uses using the EBS store, which was not available in our initial implementation. This, coupled with the redundancy offered by having consistent cloud images in multiple locations, and along with a consistent local image in our labs will significantly improve disaster recovery and the student experience.

The support from Amazon has been significant. Our AWS education grants have allowed us to extend our course without added cost. Our contact at Amazon has been very helpful in giving us the information we need to make these grants effective. However, even if the grants were not available, the costs for this course are low enough that it could be covered by the university or charged to students as a lab fee.

9. REFERENCES