Evaluation of Programmatically-Accessible Hosting Services for Visual Aids for an Online Learning Environment

Khan Academy
Mountain View, CA

Prepared by
Michael Chang
Student ID: 20332377
User ID: m9chang
3B Software Engineering

May 11, 2012

Michael Chang
30 Doncrest Rd.
Richmond Hill, ON L4B 1A2

May 11, 2012

Dr. A. Morton, Director
Software Engineering
University of Waterloo
Waterloo, ON N2L 3G1

Dear Dr. A. Morton:

I have completed my fifth work term, following my 3B term. Please find enclosed my fourth work term report entitled: " Evaluation of Programmatically-Accessible Hosting Services for Visual Aids for an Online Learning Environment " for Khan Academy, a non-profit organization with the mission of providing "a free world-class education for everyone everywhere". My manager was B. Eater, and our team was primarily responsible with creating interactive content ("exercises") on the site.

This report focuses on selecting a service for hosting images to be used in exercises on Khan Academy. This problem is one that I encountered on my work term.

I wish to thank B. Eater and B. Alpert for their advice on the technical content of this report. I also wish to thank C. Kleynhans and M. Yee for recommending the use of the uw-wkrpt LaTeXclass to format this report.

I hereby confirm that I have received no help, other than what is mentioned above, in writing this report. I also confirm that this report has not been previously submitted for academic credit at this or any other academic institution.

Sincerely,

Michael Chang
Student ID: 20332377

Executive Summary

Khan Academy provides exercises to help learners practice and review concepts that they've learned. Historically, these exercises have been built using the BSD-licensed khan-exercises framework. It is difficult to find and retain people to contribute exercises using the framework, as they both need to know how to teach and how to program in JavaScript.

Khan Academy is developing new tools to allow teachers to contribute to exercises without knowing how to program in JavaScript. These tools allow teachers to build a large bank of questions, which can be bundled together into an exercise. The initial implementation of the question editor only allowed users to embed images by hotlinking them from somewhere else.

This document describes the selection of a hosting service for the images that are uploaded in the question editor. It describes the criteria by which the services were evaluated, and uses the multi-criteria decision-making methodology to rank the services.

Table of Contents

List of Figures

List of Tables

1   Introduction

Khan Academy provides exercises to help learners practice and review concepts that they've learned. Historically, these exercises have been built using the BSD-licensed khan-exercises framework, with each problem being randomly generated. This works well for exercises that focus on computation, such as exercises for basic addition, vectors, and matrices. However, it is cumbersome for word problems in math, and even more so for other subjects, such as biology, chemistry, math, or history. Furthermore, people who wish to contribute exercises must both be proficient in teaching and in programming in JavaScript – only a small fraction of contributions received meet both content and technical standards to be incorporated into Khan Academy.

1.1   Static Questions

Khan Academy is developing new tools to allow teachers to contribute to exercises without knowing how to program in JavaScript. These tools allow teachers to build a large bank of questions, and bundle questions together into exercises. These tools will enable more teachers to contribute content to Khan Academy, which will help Khan Academy reach complete coverage of High School mathematics sooner. These tools will also enable content creators to create exercises in topics, such as World War I, that are unsuitable for the programmatic creation of question content. This report focuses on one particular component of this toolkit – the question editor.

```
![Khan Academy Logo][0]
[0]: https://khan-academy.appspot.com/images/ka-simplified-logo.png
```

Figure 1-1: Static Questions - Image Embed Syntax.

The question editor uses a combination of Markdown syntax for formatting, and LaTeX syntax for mathematical formulas. The initial version of the question editor allowed images to be inserted by using the syntax shown in figure 1-1. However, the syntax provided only allowed users to embed images by hotlinking them from another server, such as Dropbox.

1.2   Problem Statement

Hotlinking is considered poor practice for a variety of reasons. In particular, it reduces site performance for Khan Academy users, due to the need for additional DNS lookups and HTTP connections. It also puts undue financial costs on the owner of the other server. In order to remove the need for content creators to hotlink images, Khan Academy needs to procure a hosting service for the images used in the static questions.

This report seeks to select an appropriate hosting provider for images included in static questions on Khan Academy.

The Khan Academy website is primarily written in Python, and hosted on Google App Engine.

2 Analysis

First, we develop criteria and consider various hosting providers. We use the weighted sum model,[1] a method of multi-criteria decision making, to select an alternative.

2.1 Criteria

Criteria were selected based on past experience and overall organization priorities.

2.1.1 Existing Relationship With Provider

Do we already pay for services from this provider? Ranked from 1 (no prior relationship) to 5 (already use and pay for their services for mission-critical loads).

2.1.2 Team Familiarity

How much work would a developer need to learn how to use this service? Ranked from 1 (would need extensive training) to 5 (previously used in mission-critical loads).

2.1.3 User Impact

How do we think the user experience will be impacted by this alternative? Questions are served thousands of times a day, and are written to / edited infrequently. Are the URLs for accessing resources stored on this service cacheable? Ranked from 1 (need to generate a new URL for every user) to 5 (URLs can be cached for at least one year).

2.1.4 Tools

How easy is it to administrate and manage the content hosted on the service? Ranked from 1 (requires assistance from hosting provider) to 5 (well-maintained web console and

---

[1]This method is covered in MSCI 261, a core course in the software engineering curriculum at the University of Waterloo. If you need a refresher, you can read about it on Wikipedia at http://en.wikipedia.org/wiki/Weighted_sum_model.

third-party tools available).

2.1.5   Programmatic Access

How easy is it to programmatically upload content to the service? Ranked from 1 (web upload only) to 5 (documented API, with maintained libraries available, and revokable API keys for automated agents)

2.1.6   Custom ACLs Support

Ranked from 1 (binary account-wide full-access/no-access) to 5 (fine-grained per-file ACLs).

2.2   Alternatives

Possible solutions were generated by asking current employees about services they have used in the past (either professionally or personally).

2.2.1   S3

This service is provided by Amazon Web Services. S3 can be accessed using the boto library [1]. In addition to providing an API for uploads, this service supports Cross-Origin Resource Sharing (CORS) to allow for direct uploads from users[2].

2.2.2   Blobstore API (Google App Engine)

This service is provided by Google as a part of its Google App Engine offering, and targeted towards larger uploads. For Python users, the Blobstore API is accessed through a module provided with the Google App Engine SDK. End-users upload their files to an API-generated URL using a standard web form. Once the file is uploaded, Google notifies the application with a server-side callback. Blobs must be served by adding a download handler to a GAE application [3], except for images, which can be served directly using the App Engine Images API [4].

2.2.3   Cloud Storage API

This service is provided by Google, and can be accessed using the boto library [1], as well as through a model provided with the Google App Engine SDK [5]. Images can be served directly using the App Engine Images API citeref:gaeimage.

2.2.4   Imgur

This service is provided by Imgur. Commercial API access is provided through Mashape.

2.2.5   Dropbox

This service is provided by Dropbox. Users can generate URLs for files using online, mobile, and desktop clients. Pricing is per-user, and tiered based on the amount of space used.

2.3   Multi-Criteria Decision-Making

We note that the MCDM usually isn't very robust if the criteria weights are changed – in fact, it is possible to make most of the alternatives an "optimal" solution by changing the weights. However, we proceed with the weights chosen, noting that we should verify the validity of the criteria weights if we need to make a similar decision in the future.

3 Conclusions

As shown in table 3-1, Amazon's S3 best meets the criteria outlined.

Table 3-1: Multi-Criteria Decision Making Results.

Criteria	Wt.	S3		Blobstore		Storage API		Imgur		Dropbox	
Relationship	1	4	4	5	5	4	4	1	1	4	4
Familiarity	3	4	12	5	15	2	6	1	3	2	6
User Impact	3	4	12	1	3	1	3	5	15	4	12
Tools	2	4	8	4	8	5	10	1	2	2	4
APIs	2	5	10	4	8	4	8	2	4	3	6
ACLs	2	4	8	1	2	5	10	1	2	3	6
Total			**54**		**41**		**41**		**27**		**38**

4   Recommendations

We recommend that Khan Academy create an S3 bucket to store images for static questions using their existing Amazon Web Services account.

References

[1] J. Riley, M. Garnaat, T. Parslow, C. Bednarski, J. Saryerwinnie, J. Sondow, T. Grainger, D. Lindsley, and M. Stevens, *Boto*, https://github.com/boto/boto/blob/develop/README.rst, 2012.

[2] *Amazon S3 announces Cross-Origin Resource Sharing (CORS) support*, http://aws.amazon.com/about-aws/whats-new/2012/08/31/amazon-s3-announces-cross-origin-resource-sharing-CORS-support/, Amazon Web Services, Inc., 2012.

[3] *Blobstore Python API Overview*, https://developers.google.com/appengine/docs/python/blobstore/overview, Google Inc., 2013.

[4] *Functions: get_serving_url(blob_key, size=None, crop=False, secure_url=None)*, https://developers.google.com/appengine/docs/python/images/functions#Image_get_serving_url, Google Inc., 2013.

[5] *Google Cloud Storage Python API Overview*, https://developers.google.com/appengine/docs/python/googlestorage/overview, Google Inc., 2013.