Approximate Algorithmic Image Matching to Reduce Online Storage Overhead of User Submitted Images

Braden D. Licastro
Department of Computer Science
Allegheny College
licastb@allegheny.edu
http://skynetgds.no-ip.biz/srthesis

October 10, 2013

Abstract

Websites similar to Photobucket and imgur must be able to store massively increasing quantities of data; most of these sites already implement systems that limit the number, size, type of image, and compress the uploaded images to save space. Though these systems can be paired with file expiration times in which an upload is deleted, the cost of storage and backups of this data can be high. To further address this problem, the research aims to create an intelligent image uploading system capable of identifying near-duplicate image uploads on-the-fly; This system also provides the user with a higher resolution copy of their image from the server if available, thus providing better user experience while reducing unnecessary redundancy. By using the proposed system, the user will benefit from improved quality and service while the business can reduce storage and other various costs.

1 Introduction

Computers are absolutely everywhere, and society is becoming more digitized every day creating a need for storage of large amount of files. When running a website, storage space is at a premium, and the physical disks in the servers being used may not be local, but may in fact be located around the world. By allowing user submitted data, it quickly becomes apparent that data redundancy reduction can save a significant amount of storage space. This research will target the problem of data redundancy and provide a solution that will reduce not only storage costs but also inefficiencies. An implementation of a file management system will be introduced that is capable of managing large volumes of files using a database of location pointers and hash keys that identify each individual file. By using pointers, it is possible for the files to be stored on one central location or distributed across several systems while avoiding long search times. By using text based entries in a database, a hashing function can be implemented which is capable of producing a unique hash code for each file in the collection. Because this key is unique for each different file, it is a reliable gauge of uniqueness. When multiple files are found to be identical only one physical copy will be kept but pointers to the database entry will allow easy access. On large, distributed network storage servers, the benefits become ever more apparent. As users store files on the disks, the system will hash each file and search for a match in a database, thus allowing the user access to the data, but eliminating the overhead of storing numerous copies of identical data. An increase in computation time will exist but should be minimal as text comparisons will be the primary task in finding uniqueness, but this will be outweighed by minimized physical storage requirements and costs of data backup. The implementation of this system could be used in tandem with other currently implemented methods of file space cost reduction technologies out of the box. By not only using the proposed method of space requirement reduction, it could in theory be possible that with data expiration times, file size maximums, and similar technologies, that storage needs will plateau after the initial surge of additions and the expiration time has passed for the earliest uploaded file. This would allow webmasters to better predict overall needs and better predict upload trends and react accordingly with preventative maintenance and any required space additions.

2 Related Work

Summarize the previously published papers and books that are related to your proposed research. Whenever possible, you should compare and contrast your approach with the ones that have been discussed in the past. As you describe your papers, please make sure that you cite them properly. Let's use the rest of this section to generate out bibliography! [8] [1] [6] [2] [3] [9] [5] [4] [10] [7].

3 Method of Approach

Use technical diagrams, equations, algorithms, and paragraphs of text to describe the research that you intend to complete. See the LaTeX source file for the proposal to learn how Figure 1 and Table 1 were included. Be sure to number all figures and tables and to explicitly refer to them in your text.

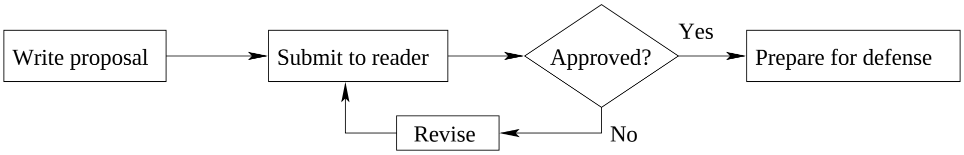

Figure 1: Flow graph for proposal-writing

Task	Begin Date	End Date
First draft	Now	20 Sept
Second draft	20 Sept	27 Sept
Third draft	27 Sept	4 Oct
Fourth draft	4 Oct	11 Oct
Fifth draft	11 Oct	18 Oct

Table 1: Proposed work schedule

4 Evaluation Strategy

Explain what steps you will take to evaluate your proposed method. If you intend to conduct experiments, then you must clearly define your evaluation metrics.

5 Research Schedule

Identify the main phases and tasks of your research project and set deadlines for when you will be able to complete each of these items.

6 Conclusion

Provide a summary of your proposed research and suggest the impact that it may have on the discipline of computer science. If possible, you may also suggest some areas for future research.

References

[1] Angela Bradley. Renaming PHP Uploads. http://php.about.com/od/advancedphp/ss/rename_upload.htm, 2011. [Online; accessed 10-October-2013].

[2] CatPa.ws. PHP GD Duplicate Image Finder. http://www.catpa.ws/php-duplicate-image-finder/, 2010. [Online; accessed 13-September-2013].

[3] Ubuntu Documentation. ApacheMySQLPHP - LAMP Server Setup Guide. http://php.net/manual/en/ref.image.php, 2013. [Online; accessed 21-August-2013].

[4] Jun Jie Foo, Justin Zobel, Ranjan Sinha, and S. M. M. Tahaghoghi. Detection of Near-duplicate Images for Web Search. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, CIVR '07, pages 557–564, New York, NY, USA, 2007. ACM.

[5] The PHP Group. GD and Image Functions. http://php.net/manual/en/ref.image.php, 2013. [Online; accessed 13-October-2013].

[6] The PHP Group. POST Method Uploads. http://de.php.net/manual/en/features.file-upload.post-method.php, 2013. [Online; accessed 09-September-2013].

[7] David C. Lee, Qifa Ke, and Michael Isard. Partition Min-hash for Partial Duplicate Image Discovery. In *Proceedings of the 11th European Conference on Computer Vision: Part I*, ECCV'10, pages 648–662, Berlin, Heidelberg, 2010. Springer-Verlag.

[8] Dejan Marjanovic. PDO vs. MySQLi: Which Should You Use? http://net.tutsplus.com/tutorials/php/pdo-vs-mysqli-which-should-you-use/, 2012. [Online; accessed 02-October-2013].

[9] nixCraft. Ubuntu Linux: Install or Add PHP-GD Support to Apache Web Server. http://www.cyberciti.biz/faq/ubuntu-linux-install-or-add-php-gd-support-to-apache/, 2012. [Online; accessed 30-September-2013].

[10] S H. Srinivasan and Neela Sawant. Finding Near-duplicate Images on the Web Using Fingerprints. In *Proceedings of the 16th ACM International Conference on Multimedia*, MM '08, pages 881–884, New York, NY, USA, 2008. ACM.