SPRINGER BRIEFS IN COMPUTER SCIENCE

Hala ElAarag

Web Proxy Cache Replacement Strategies

Simulation, Implementation, and Performance Evaluation

Springer

SpringerBriefs in Computer Science

Series Editors

Stan Zdonik
Peng Ning
Shashi Shekhar
Jonathan Katz
Xindong Wu
Lakhmi C. Jain
David Padua
Xuemin Shen
Borko Furht
V. S. Subrahmanian
Martial Hebert
Katsushi Ikeuchi
Bruno Siciliano

For further volumes:
http://www.springer.com/series/10028

Hala ElAarag

Web Proxy Cache Replacement Strategies

Simulation, Implementation, and Performance Evaluation

With Contributions by Sam Romano and Jake Cobb

Springer

Hala ElAarag
Department of Mathematics
and Computer Science
Stetson University
DeLand, FL
USA

ISSN 2191-5768 ISSN 2191-5776 (electronic)
ISBN 978-1-4471-4892-0 ISBN 978-1-4471-4893-7 (eBook)
DOI 10.1007/978-1-4471-4893-7
Springer London Heidelberg New York Dordrecht

Library of Congress Control Number: 2012954855

© Hala ElAarag 2013
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To my loving parents, husband, and children

Preface

The need for this book stems from the sheer amount of modern web traffic coupled with increases in cacheable, bandwidth-consuming multimedia objects. In this book, we provide the most comprehensive study for proxy cache replacement strategies. We categorize these strategies into four categories; recency-based, frequency-based, recency/frequency-based, and function-based. We provide a quantitative comparison of cache replacement strategies on the category level and then compare the best strategies of each category based on very important performance metrics. We then diverge from these replacement policies by constructing a model, represented in the weights and structure of a neural network, from actual web traffic. This approach has the advantage of incorporating subtle traffic pattern information which may be difficult for a human to discern when designing a top-down algorithm. Furthermore, we provide a single method which is capable of creating multiple models; this allows models to be created which target localized traffic patterns as well as general ones. We first provide the simulation architecture, setup, parameters, and results of this novel technique then we explain the implementation details in the Squid proxy server.

This book is based on a number of my publications co-authored with my students Sam Romano, now a software engineer at Google and Jake Cobb, now a Ph.D. student at Georgia Institute of Technology. I would like to thank them both for their contributions to this research. I would like to thank the anonymous reviewers of *Simulation: Transactions of the Society for Modeling and Simulation International*, Sage Publication, *The Journal of Systems and Software*, Elsevier, and *Neural Computing and Applications*, Springer-Verlag, for their comments and critique that helped to improve the quality of this research. I would also like to thank the staff members of Springer, Wayne Wheeler for introducing me to the Springer Brief series and Simon Rees, for the follow up through the preparation of this book.

Last, but not least, I am deeply grateful to my family for their continuous support and encouragement.

Contents

Chapter 1
Introduction

Keywords Web proxy servers · Proxy cache · Cache replacement strategies · Neural networks · Squid proxy server

With the debut of Web 2.0, many researchers are interested in studying their workload characteristics in order to design more efficient servers. YouTube is an example of a very popular Web 2.0 site, and hence, it was the choice for many to conduct such research [1]. Gill et al. [1] conducted an extensive analysis of YouTube workload and observed 25-million YouTube transactions over a three-month period that included the downloads of 600,000 videos. They compared traditional Web workload characteristics to that of YouTube and found many similarities. They concluded that traffic characterization of YouTube suggests that caching popular video files on Web proxy servers reduces network traffic and increases scalability of YouTube servers [1].

User-perceived delay results from both overload of individual servers and network congestion. Proxy caches are used to address both issues by attempting to serve requests locally. There are several decisions that must be made such as cache placement and replacement. Our main focus will be the cache replacement problem, the process of evicting objects in the cache to make room for new objects, applied to Web proxy servers. Proxy servers want to serve as many objects from the cache as possible, serve as much data from the cache as possible, or both. Optimizing both is ideal, but many practical algorithms optimize for one over the other.

There are many replacement strategies to consider when designing a proxy server. The most commonly known cache replacement strategies are least frequently used (LFU) and least recently used (LRU). Podlipnig et al. [2] provided a survey of Web cache replacement strategies. They have done well to not only list well-known strategies, but also categorize the strategies into five groups.

Although both a survey and classification of these strategies are available, there is not a known record of results comparing the majority of the cache replacement

H. ElAarag, *Web Proxy Cache Replacement Strategies*,
SpringerBriefs in Computer Science, DOI: 10.1007/978-1-4471-4893-7_1,

© Hala ElAarag 2013

algorithms. Balamash and Kunz [3] compared cache replacement policies qualitatively rather than quantitatively. Many of the works consulted for this research presented results for different strategies against, at most, three to five other strategies. However, their results cannot be compared effectively because most literature devised their experiments with differences in their implementations, such as the use of auxiliary caching, representation of Web object characteristics. There is also the difference in trace files between experiments and a large range of different sizes used for the cache space. Lastly, different proposals used different criteria on when to cache an object such as ignoring *PHP* files, *cgi-bin* scripts, POST requests, and simply just using all requests presented in a particular trace file.

In spite of the many cache replacement policies proposed in the past years, the most popular Web proxy software, Squid, uses least recently used as its default strategy. We believe that this is because there has not been a comprehensive study presented that compares these strategies quantitatively. As Wong [4] mentions "all policies were purported to perform better than others, creating confusion as to which policy should be used".

In this book, we present a study of cache replacement strategies designed for static Web content, as opposed to dynamic Web content seen in some Web 2.0 applications. Most caching that occurs with dynamic content occurs on the browser side and does not occur from the standpoint of proxy servers. As a result, these strategies have not been considered. We have researched how proxy servers can still improve performance by caching static Web content such as cascading style sheets, java script source files, and most importantly larger files such as images (jpeg, gif, etc.).

This topic is particularly important in wireless ad hoc networks. In such networks, mobile devices act as proxy servers for a group of other mobile devices. However, they have limited storage space. The extensive research on cache replacement policies we provide in this book is crucial for such environments with small cache sizes and limited battery life.

Neural networks have been employed in a number of applications, particularly in the area of pattern recognition. Neural networks are able to learn by experience and hold an internal representation of meaning in data. An appropriately structured neural network will be able to generalize the knowledge acquired from training to data that lies outside the training set. This property makes neural networks useful for solving problems that contain uncertainty or have a problem space which is too large for an exhaustive search. We use neural networks to solve the Web proxy cache replacement problem. A bottom-up approach is justified by the heavy variability in Web traffic which makes general characterization of that traffic difficult. Neural networks are selected for their strength in pattern recognition in noisy data sets. Additionally, they can learn from example by training against a data set, yet are able to generalize beyond the training set. Thus, they are well suited for developing a general replacement strategy from a set of data samples. This approach has the advantage of incorporating subtle traffic pattern information which may be difficult for a human to discern when designing a top-down algorithm.

The rest of the book is organized as follows. Chapter 2 defines Web requests and the characteristics of Web objects and presents some background information about Web proxy servers and Squid. Chapter 3 presents some background information about artificial neural networks. In chap. 4, we present *TWENTY SEVEN* cache replacement strategies we simulated against different performance metrics [5]. To the best our knowledge, this is the most comprehensive quantitative analysis of Web cache replacement strategies. Chapter 5 presents our novel approach to Web proxy cache replacement that uses neural networks for decision making and evaluates its performance and decision structures [6]. We finally present in chap. 6 the implementation of our neural network proxy cache replacement scheme in a real environment, namely in the Squid proxy server [7].

References

1. P. Gill et al., *Youtube traffic characterization: a view from the edge, Proceedings of the 7th ACM SIGCOMM conference on internet measurement* (San Diego, California, USA, 2007), pp. 15–28
2. S. Podlipnig, L. Boszormenyi, A survey of web cache replacement strategies. ACM Comput. Surveys **35**(4), 374–398 (2003)
3. A. Balamash, M. Krunz, An overview of web caching replacement algorithms. IEEE Commun. Surveys and Tutorials **6**(2), 44–56 (2004)
4. K. Wong, Web cache replacement policies: a pragmatic approach. IEEE Network **20**(1), 28–34 (2006)
5. H. ElAarag, S. Romano, A quantitative study of web cache replacement strategies using simulation, simulation: transactions of the society for modeling and simulation international, Sage Publication, Published online before print July 25, 2011. 88(5), 507–541 (2012). doi: 10.1177/0037549711414152
6. W. Cobb, H. ElAarag, Web proxy cache replacement scheme based on back propagation neural network. J. Syst. Software **81**(9), 1539–1558 (2008). doi:10.1016/j.jss.2007.10.024
7. S. Romano, H. ElAarag, A neural network proxy cache replacement strategy and its implementation in the squid proxy server. Neural Comput. Appl **20**(1), 59–78 (2011). doi:10.1007/s00521-010-0442-0