Reducing Network Traffic

Web Caching

O'REILLY®

Duane Wessels

Web Caching

Duane Wessels

O'REILLY®
Beijing · Cambridge · Farnham · Köln · Paris · Sebastopol · Taipei · Tokyo

Web Caching
by Duane Wessels

Published by O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472.

Editors: Nathan Torkington and Paula Ferguson

Production Editor: Leanne Clarke Soylemez

Cover Designer: Edie Freedman

Printing History:

June 2001: First Edition.

Library of Congress Cataloging-in-Publication Data

Wessels, Duane.
Web Caching/Duane Wessels
p. cm.
ISBN 1-56592-536-X
1. Cache memory. 2. Browsers (Computer programs) 3. Software configuration management. 4. World Wide Web. I. Title.

TK7895.M4 W45 2001
004.5'3--dc21 2001033173

ISBN: 1-56592-536-X

[C]

Preface .....

1. Introdu
1.1 We
1.2 We
1.3 Wh
1.4 Wh
1.5 Typ
1.6 Cac
1.7 Me
1.8 Pro

2. How W
2.1 HT
2.2 Is I
2.3 Hit
2.4 Hit
2.5 Val
2.6 For
2.7 Cac

3. Politics
3.1 Pri
3.2 Rec
3.3 Co

2.4 Hit Ratios

How can we measure the effectiveness of our caches? One such measurement is called the *cache hit ratio.* This is the percentage of requests that are satisfied as cache hits. Usually, this includes both validated and unvalidated hits. Validated hits can be tricky because these requests are forwarded to origin servers, incurring slight bandwidth and latency penalties. Note that the cache hit ratio tells you only how many requests are hits—it doesn't tell you how much bandwidth or latency has been saved.

The measurement that does tell you about bandwidth is called the *byte hit ratio.* Instead of counting only requests, this measure is based on the number of bytes transferred. Cache hits for large objects contribute more to the byte hit ratio than do small objects. The byte hit ratio measures how much bandwidth your cache has saved, but there are different ways to calculate it.

One way is to compare the sum of object sizes for cache hits and cache misses. However, this technique has a couple of shortcomings. For example, it doesn't include request traffic. Counting the request traffic probably doesn't matter much because it's relatively small, and most of the data flows in the other direction (into your network, not out of it). This technique might not count the small 304 (Not Modified) responses either. However, the bigger problem is in accounting for requests aborted by the user. Consider a cache miss for a 100 KB object. If the cache downloads the entire response, but the user aborts the transfer at 50 KB, we used more server-side bandwidth than on the client-side. If we instead count bytes transferred on the network, we'll get a more accurate figure for bandwidth savings.

What sort of cache hit ratio values can you expect to achieve? Any reasonably-sized cache should be able to reach 30%. Some of the largest and busiest caches deployed today can make it as high as 70%. Byte hit ratios are normally less than cache hit ratios, often by as much as 10%. That is, a 50% cache hit ratio usually corresponds to a 40% byte hit ratio. We'll see the reason for this in Section A.1, "Reply and Object Sizes." Small objects, such as images and HTML pages, tend to have more cache hits than large objects such as audio and PostScript files.

Three factors affect hit ratios: cache size, number of clients, and freshness heuristics. Obviously, a larger cache holds more objects. The chance that any given request is a cache hit increases as the cache size increases, but the relationship is not linear. In other words, doubling your cache size does not double your hit ratio. Research has shown that the relationship is pseudologarithmic [Duska, Marwood and Feely, 1997]. I show the same in Appendix A with cache simulations. Even if your cache size is infinite, there is an upper limit on the hit ratio you can achieve. Section 10.2, "Disk Space," explains how you can determine an appropriate cache size for your installation.