

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

Adapting ACME to the Database Caching Environment

**A thesis presented in partial
fulfilment of the requirements for the degree of**

Master of Science

in

Information Systems

at Massey University, Palmerston North, New Zealand.

Faizal Riaz-ud-Din

2003

Abstract

The field of database cache replacement has seen a great many replacement policies presented in the past few years. As the challenge to find the optimal replacement policy continues, new methods and techniques of determining cache victims have been proposed, with some methods having a greater effect on results than others. Adaptive algorithms attempt to adapt to changing patterns of data access by combining the benefits of other existing algorithms. Such adaptive algorithms have recently been proposed in the web-caching environment. However, there is a lack of such research in the area of database caching. This thesis investigates an attempt to adapt a recently proposed adaptive caching algorithm in the area of web-caching, known as Adaptive Caching with Multiple Experts (ACME), to the database environment. Recently proposed replacement policies are integrated into ACME's existing policy pool, in an attempt to gauge its ability and robustness to readily incorporate new algorithms. The results suggest that ACME is indeed well-suited to the database environment, and performs as well as the best currently caching policy within its policy pool at any particular point in time in its request stream. Although execution time increases by integrating more policies into ACME, the overall time saved increases by avoiding disk reads due to higher hit rates and fewer misses on the cache.

Acknowledgements

In the Name of Allah, The Most Beneficent, The Most Merciful

Praise be to Allah, The One and Only True God, who has blessed me with the opportunity to finally achieve my long-standing ambition to undertake my Masters degree.

I am immensely grateful to Markus Kirchberg, for agreeing to supervise me, and for his comments and invaluable criticism during the course of this dissertation. I have greatly appreciated his on-going commitment, dedication and patience with helping me to complete this project. This thesis would not have been possible without his contribution.

I am also grateful to Ismail Ari for providing the source code and traces for ACME, and for providing helpful comments. I would like to acknowledge Xiaodong Zhang, Song Jiang, Sam H. Noh, and Heung Seok Jeon for providing me with the database traces.

I would also like to acknowledge the financial support given to me by the Information Science Research Centre at Massey University.

I would like to thank my parents for their encouragement and support, and for giving me the opportunity to study through to university.

Last, but by no means least, I would like thank my dear wife Leah, for providing me with endless hours of encouragement and support. Thank you for all those hours you spent patiently sitting by me, with motivating me, with your helpful criticism, and with the final documentation. This thesis would not have been what it is without your input.

Table of Contents

ABSTRACT	I
ACKNOWLEDGEMENTS	II
TABLE OF CONTENTS	III
LIST OF FIGURES AND TABLES	V
1. INTRODUCTION	1
1.1 CACHE REPLACEMENT IN DATABASES	1
1.2 BENEFITS AND DRAWBACKS OF CACHE REPLACEMENT POLICIES	4
1.3 CACHE REPLACEMENT IN OPERATING SYSTEMS AND WEB SERVERS.....	5
1.4 MOTIVATION	6
1.5 PRESENTATION OF THE REMAINDER OF THIS DISSERTATION.....	8
2. RELATED WORK	9
2.1 ADAPTIVE ALGORITHMS	9
2.2 ACME: ADAPTIVE CACHING WITH MULTIPLE EXPERTS	9
2.2.1 <i>Description of ACME</i>	9
2.2.2 <i>Benefits of ACME</i>	12
3. ADAPTING ACME TO THE DATABASE ENVIRONMENT	14
3.1 DEFINING THE INTENDED DATABASE ENVIRONMENT	14
3.2 ALLOWING FOR SAME SIZED PAGES	15
3.3 REMOVAL OF WEB-SPECIFIC POLICIES.....	15
3.4 ADDITIONAL DEMANDS OF DATABASE SPECIFIC POLICIES.....	15
4. AN IMPLEMENTATION OF THE DATABASE-ADAPTED ACME ALGORITHM	17
4.1 RELEASE MECHANISMS.....	17
4.2 CHOICE OF POLICIES.....	17
4.3 REVIEW OF POLICIES	18
4.3.1 <i>Existing policies</i>	18
4.3.1.1 <i>Random</i>	18
4.3.1.2 <i>FIFO</i>	19
4.3.1.3 <i>LIFO</i>	19
4.3.1.4 <i>LRU</i>	19
4.3.1.5 <i>LFU</i>	19
4.3.1.6 <i>LFUDA</i>	20
4.3.1.7 <i>MFU</i>	20
4.3.1.8 <i>MRU</i>	20

4.3.2 <i>New policies</i>	21
4.3.2.1 <i>LIRS</i>	21
4.3.2.2 <i>LRFU</i>	23
4.3.2.3 <i>LRU-K</i>	23
4.3.2.4 <i>SFIFO</i>	24
4.3.2.5 <i>2Q</i>	25
4.3.2.6 <i>W²R</i>	27
4.4 CLASSIFICATION OF POLICIES	28
5. EXPERIMENTAL RESULTS	29
5.1 TRACES	29
5.2 EXPERIMENTAL METHODOLOGY	30
5.2.1 <i>Combined effect of all policies on the Real Cache</i>	31
5.2.2 <i>Real Cache adaptation to the current best policy</i>	31
5.2.3 <i>Effect of the weaker policies on the Real Cache</i>	31
5.2.4 <i>The adaptive nature of database-adapted ACME</i>	31
5.2.5 <i>Investigation of susceptibilities to well-known weaknesses</i>	32
5.2.6 <i>Average time to select a victim</i>	32
5.2.7 <i>Average time loss for each additional policy</i>	32
5.2.8 <i>Switching of current best policies</i>	32
5.2.9 <i>The effect of disk reads on the total execution time</i>	33
5.3 ANALYSIS AND DISCUSSION	33
5.3.1 <i>Relative performance of policies</i>	33
5.3.2 <i>The performance of the Real Cache</i>	35
5.3.3 <i>The adaptive behaviour of ACME</i>	38
5.3.4 <i>The effect of having different cache sizes</i>	40
5.3.5 <i>Effect of replacement policies on time performance</i>	46
5.3.6 <i>Machine learning takes time to take effect</i>	47
5.3.7 <i>Effect on all policies by introducing well-known susceptibilities</i>	49
5.3.8 <i>The effect of misses on the total processing time</i>	53
5.3.9 <i>Flaw in the original ACME implementation</i>	55
5.3.10 <i>Outcomes of research</i>	57
6. FUTURE PROSPECTS	58
7. CONCLUSIONS	60
8. REFERENCES	61

List of Figures and Tables

Figures

Figure 1 - Design of Adaptive Caching with Multiple Experts (ACME).....	10
Figure 2 - Adaptive caching	12
Figure 3 - Requests vs Hit rates (all policies), DB2 Trace	34
Figure 4 - Requests vs Hit rates (all policies), OLTP Trace.....	34
Figure 5 - Requests vs Hit rates (2Q and FIFO only), DB2 Trace.....	36
Figure 6 - Requests vs Hit rates (2Q and FIFO only), OLTP Trace.....	36
Figure 7 - Real Caches for different policy combinations, DB2 Trace.....	37
Figure 8 - Real Caches for different policy combinations, OLTP Trace.....	38
Figure 9 - Request vs Hit rate (2Q and LRU only), Synthetic Trace A.....	39
Figure 10 - Request vs Hit rate (LRU-2 and LRU only), OLTP Trace	40
Figure 11 - Cache Size vs Hit rate (all policies), DB2 Trace.....	41
Figure 12 - Cache Size vs Hit rate (all policies), OLTP Trace	41
Figure 13 - The effect of cache size on the Real Cache hit rate.....	42
Figure 14 - The effect of cache size on the time taken to select a victim.....	44
Figure 15 - The increase in time by adding each policy to the policy pool	46
Figure 16 - The gradual adaptation of the Real Cache to the best policy, DB2 trace.....	48
Figure 17 - The effect of sequential flooding on the policy pool.....	49
Figure 18 - The effect of sequential scans on the policy pool	50
Figure 19 - The effect of skewed high reference frequencies on the policy pool	51
Figure 20 - Flawed results showing Request vs Hit rate (2Q and LIRS only), DB2 Trace	56
Figure 21 - Corrected results showing Requests vs Hit rate (2Q and LIRS only), DB2 Trace	56

Tables

Table 1 - Classification of policies used in database-adapted ACME.....	28
Table 2 - Real Cache hit rates for different cache sizes	42
Table 3 - Average time to select a victim for different cache sizes.....	44
Table 4 - Execution times with and without disk reads for 2Q and SFIFO, Synthetic Trace E.....	53
Table 5 - Execution times with and without disk reads for 2Q and SFIFO, DB2 Trace	54