ANALYSIS OF CACHE PERFORMANCE FOR OPERATING SYSTEMS AND MULTIPROGRAMMING

THE KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE

PARALLEL PROCESSING AND FIFTH GENERATION COMPUTING

Consulting Editor
Doug DeGroot

Other books in the series:

PARALLEL EXECUTION OF LOGIC PROGRAMS,
John S. Conery ISBN 0-89838-194-0

PARALLEL COMPUTATION AND COMPUTERS FOR ARTIFICIAL INTELLIGENCE
Janusz S. Kowalik ISBN 0-89838-227-0

MEMORY STORAGE PATTERNS IN PARALLEL PROCESSING
Mary E. Mace ISBN 0-89838-239-4

SUPERCOMPUTER ARCHITECTURE
Paul B. Schneck ISBN 0-89838-234-4

ASSIGNMENT PROBLEMS IN PARALLEL AND DISTRIBUTED COMPUTING
Shahid H. Bokhari ISBN 0-89838-240-8

MEMORY PERFORMANCE OF PROLOG ARCHITECTURES
Evan Tick ISBN 0-89838-254-8

DATABASE MACHINES AND KNOWLEDGE BASE MACHINES
Masaru Kitsuregawa ISBN 0-89838-257-2

PARALLEL PROGRAMMING AND COMPILERS
Constantine D. Polychronopoulos ISBN 0-89838-288-2

DEPENDENCE ANALYSIS FOR SUPERCOMPUTING
Utpal Banerjee ISBN 0-89838-289-0

DATA ORGANIZATION IN PARALLEL COMPUTERS
H.A.G. Wijshoff ISBN 0-89838-304-8

ANALYSIS OF CACHE PERFORMANCE FOR OPERATING SYSTEMS AND MULTIPROGRAMMING

by

Anant Agarwal
Massachusetts Institute of Technology

with a foreword by

John L. Hennessey

KLUWER ACADEMIC PUBLISHERS
Boston/Dordrecht/London

Distributors for North America:
Kluwer Academic Publishers
101 Philip Drive
Assinippi Park
Norwell, Massachusetts 02061, USA

Distributors for the UK and Ireland:
Kluwer Academic Publishers
Falcon House, Queen Square
Lancaster LA1 1RN, UNITED KINGDOM

Distributors for all other countries:
Kluwer Academic Publishers Group
Distribution Centre
Post Office Box 322
3300 AH Dordrecht, THE NETHERLANDS

Library of Congress Cataloging-in-Publication Data

Agarwal, Anant.
Analysis of cache performance for operating systems and multiprogramming / by Anant Agarwal.
p. cm. — (The Kluwer international series in engineering and computer science ; SECS 69)
Bibliography: p.
Includes index.

ISBN-13: 978-1-4612-8897-8 e-ISBN-13: 978-1-4613-1623-7
DOI: 10.1007/978-1-4613-1623-7
1. Cache memory—Evaluation. 2. Operating systems (Computers) 3. Multiprogramming (Electronic computers) I. Title. II. Series.
TK7895.M4A33 1988
005.4'3—dc19 88-36500
CIP

Contents

List of Figures

List of Tables

Foreword

As we continue to build faster and faster computers, their performance is becoming increasingly dependent on the memory hierarchy. Both the clock speed of the machine and its throughput per clock depend heavily on the memory hierarchy. The time to complete a cache access is often the factor that determines the cycle time. The effectiveness of the hierarchy in keeping the average cost of a reference down has a major impact on how close the sustained performance is to the peak performance. Small changes in the performance of the memory hierarchy cause large changes in overall system performance. The strong growth of RISC machines, whose performance is more tightly coupled to the memory hierarchy, has created increasing demand for high performance memory systems. This trend is likely to accelerate: the improvements in main memory performance will be small compared to the improvements in processor performance. This difference will lead to an increasing gap between processor cycle time and main memory access time. This gap must be closed by improving the memory hierarchy.

Computer architects have attacked this gap by designing machines with cache sizes an order of magnitude larger than those appearing five years ago. Microprocessor-based RISC systems now have caches that rival the size of those in mainframes and supercomputers. These large cache sizes and the need for more accurate simulation have created the demand for new and more ambitious studies of cache performance. Earlier studies of caches used programs that are much smaller than the programs run today. Those earlier studies cannot adequately predict the performance of very large caches, since the benchmarks essentially fit within the cache.

Since small changes in miss rate cause large changes in system performance, our cache performance prediction must be more accurate than ever before. This requirement for accuracy means that factors that were previously ignored or estimated must be carefully analyzed and measured. A primary example of such a factor is the effect of the operating system's references on cache performance of a user program. The emergence of shared-memory multiprocessors has created another demand for accurate modeling of cache performance. These machines are extremely sensitive to cache misses because such misses cause contention for a shared resource (the bus and memory). In addition, multiprocessors cannot be accurately simulated with simple uniprocessor cache traces, due to the presence of invalidation requests and misses arising from those requests, both of which are absent in a uniprocessor trace.

This work addresses these problems by examining the performance of large

programs running in large caches and measuring an entire range of effects not examined in earlier work. To conduct these experiments the first challenge was to devise a method for collecting traces of large programs, including the operating system and multiprogramming effects that were neglected in earlier studies. A system for collecting such traces was created and is described herein.

Because of the large domain of possible cache designs and the length of the traces needed to simulate them accurately, analytical modeling techniques that can explore a range of cache organizations quickly are important. These techniques allow a designer to concentrate on a small number of alternatives out of the vast range of possible cache designs. This small number of designs can then be fully simulated to obtain accurate performance estimates.

Finally, exploring the effect of multiprocessing required obtaining traces from a multiprocessor and then constructing accurate simulation models. Since the effects of multiprocessing can have a significant impact on cache performance and an enormous impact on system performance, it is vital to obtain these traces and model these effects accurately.

The research results described here are an important resource for all computer architects. The increasing importance of the memory hierarchy in determining the performance of high speed machines and multiprocessors makes this type of exploration and analysis invaluable if our computer systems are to continue to grow in performance.

John L. Hennessy
Stanford, California

Preface

Advances in high-performance processors continue to create an increased need for memory bandwidth. Migration to multiprocessor architectures exacerbates this need causing the well known Von Neumann bottleneck to main memory. Caches can provide this bandwidth cost-effectively. However, minimizing the performance loss due to caching requires that our analysis and prediction of cache performance become more exact. Although previous studies have shown that operating system and multiprogramming activities affect the cache performance, those studies did not deal with these issues in detail, nor did they address multiprocessor cache performance, largely because of the unavailability of efficient analysis techniques and the difficulty in collecting data for these analyses. To obtain the higher hit rates needed to sustain the effectiveness of caches, we must address these issues completely.

This book investigates the performance of large caches for realistic operating system and multiprogramming workloads. Cache analysis of bus-based multiprocessors is also presented. A suite of efficient and accurate cache evaluation techniques is developed. These include: a mathematical cache model, a trace sampling and a trace stitching procedure, and a trace compaction method. The analyses use a data collection technique called ATUM to obtain realistic system traces of multitasking workloads with little distortion. An extension of ATUM that traces multiprocessors is also described.

Accurately characterizing cache behavior using ATUM traces shows that both operating system and multiprogramming activities significantly degrade cache performance, with an even greater proportional impact on large caches. Multiprocessor interference further reduces cache performance. From a careful analysis of the causes of this degradation, we explore various techniques to reduce this loss. While seemingly little can be done to mitigate the effect of system references, multitasking cache misses can be reduced with little effort. The impact of process migration, virtual versus physical addressing, and the effect of synchronization on large write-back caches in multiprocessors is investigated. We also demonstrate how analytical cache modeling, and trace sampling – with a new approach to cold-start and warm-start analysis – can be used to make large cache studies insightful and efficient.

This book is largely based on my Ph.D. thesis submitted in May 1987 at Stanford University. Chapter 7 contains new material on multiprocessor caches extending the earlier work on uniprocessor cache behavior and represents recent work done at Stanford and also at M.I.T. in collaboration with Dick Sites at Digital Equipment Corporation, Hudson, Massachusetts.

Acknowledgments

I wish to thank Professor John Hennessy for his support, guidance and encouragement, and most of all, for his enthusiasm, throughout the course of my research. Even in his terribly busy moments, he would be happily willing to discuss problems. I am indebted to Professor Mark Horowitz, who was a source of innumerable suggestions and ideas, and to Dick Sites, who was instrumental in getting me going in my research. I am also indebted to Professors Thomas Kailath, Robert Matthews and John Newkirk, who offered me the opportunity for graduate studies at Stanford.

My gratitude to the MIPS-X crew, including Arturo Salz, Paul Chow, Malcolm Wing, Scott McFarling, C. Y. Chu, Steve Richardson, Don Stark, Rich Simoni, John Acken, Steve Przybylski, Peter Steenkiste, Steve Tjiang, and Glenn Gulak, for their suggestions and criticisms, for asking all those questions that motivated my research, and for simply making my experience at Stanford a very enjoyable one.

I must not fail to acknowledge my Indian Institute of Technology clan in the valley, had it not been for whom, I would have completed my thesis far sooner, but with far more of my sanity lost. I am also grateful to Madan Valluri, Sailesh Rao, Barbara and Steve Mckee, and Yvonne and Jim Weiberg for their friendship and affection when I first came to this country.

My research was supported in part by Defense Advanced Research Projects Agency under contract # MDA903-83-C-0335 at Stanford, and under contract # N00014-87-K-0825 at MIT during the preparation of this book. I am grateful to Digital Equipment Corporation, Hudson, for making trace data available for this research.

My wife Anu was a constant source of moral support and encouragement throughout my graduate study, going through some stressful times very patiently and happily.

Finally, I dedicate this book to my parents, for their love, care and sacrifice.

Anant Agarwal
Cambridge, Massachusetts

ANALYSIS OF CACHE PERFORMANCE FOR OPERATING SYSTEMS AND MULTIPROGRAMMING