

Application of Restructuring Techniques
to the Optimization of Program Behavior
in Virtual Memory Systems

By

Jehan Francois Paris

Engineer (Free University of Brussels) 1970

Grad. (University Faculties of Our Lady of Peace, Belgium) 1972

Grad. (University of Paris VI) 1974

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Engineering

in the

GRADUATE DIVISION

OF THE

UNIVERSITY OF CALIFORNIA, BERKELEY

Approved:

[Signature] *May 18, 1981*
Chairman Date
[Signature]
D. Billinger

.....

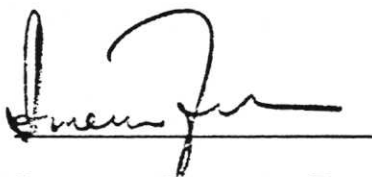
Application of Restructuring Techniques to the
Optimization of Virtual Memory Systems

Copyright © 1981 by Jehan-François Paris

Application of Restructuring Techniques to the Optimization
of Virtual Memory Systems

Jehan-François Paris

Ph. D. in Engineering
Electrical Engineering and Computer Science



Professor Domenico Ferrari
Chairman of Thesis Committee

Abstract

One of the most effective ways of obtaining a better performance from virtual memory systems consists of improving the behavior of programs in such environments. Program restructuring attempts to achieve this goal by rearranging the block-to-page mappings of programs.

The best existing restructuring algorithms take into account the *dynamic* behavior of the program to be restructured and attempt to minimize either its page fault frequency or its mean memory occupancy, which are both partial indicators of program performance.

In this thesis, we present a new class of restructuring algorithms that attempt to minimize a global index of program performance, namely its *space-time product*. The primary motivation of these algorithms is to avoid situations where a significant improvement of one index of program performance would be accompanied by a comparably sized deterioration of another index. Hence the name of "Balanced Algorithms" given to our algorithms.

Balanced Algorithms essentially attempt to minimize a restructuring-time estimate of the space-time product of the restructured program. Since they follow a common scheme, they can be easily tailored to a wide range of variable-space memory policies, including Working Set, Sampled Working Set, Global LRU and Page Fault Frequency.

We prove that BWS, the balanced algorithm tailored to Working Set environments, effectively minimizes a linear combination of the number of page faults and of the mean memory occupancy of all programs whose behavior can be described by a chain having a steady-state solution and which have at most two blocks per page.

In order to evaluate the performance of balanced algorithms under various memory policies and to compare it to those of other restructuring algorithms, a series of trace-driven experiments simulating the behavior of programs before and after restructuring were conducted. These simulations show that BWS, the balanced algorithm tailored to Working Set environments, performs significantly

better than the two best existing restructuring algorithms. Similar results were found with the balanced algorithm tailored to Sampled Working Set environments. BPSI, the balanced algorithm tailored to Global LRU environments, exhibited only a marginal superiority over its rivals, while no clear winner emerged for the Page Fault Frequency environments.

Another consequence of our choice of a global indicator of program performance as restructuring criterion is to allow our approach to be extended to *segmentation environments*, or which no efficient restructuring algorithms have been proposed.

Here too, we prove that BTWWS, the balanced algorithm tailored to Time-Window Working Set environments, minimizes a linear combination of the number of segment faults and of the mean memory occupancy of all programs whose behavior can be described by a chain having a steady-state solution and which have at most two blocks per segment.

Experimental evidence is also presented showing that BTWWS can significantly improve the segment fault frequency of a program without causing any comparable increase of its memory occupancy. On the other hand, our simulations indicate that BSFF, the balanced algorithm tailored to Segment Fault Frequency environments, does not bring any improvement to either indices of program performance.

TABLE OF CONTENTS

Acknowledgements	iii
1. Virtual Memory Concepts	1
1.1. Introduction	1
1.2. Virtual Memory Management	1
1.2.1 Paging v. Segmentation	2
1.2.1.1. Paged Virtual Memory	2
1.2.1.2. Purely Segmented Virtual Memory	2
1.2.1.3. Paged and Segmented Virtual Memory	3
1.2.2 Fetch Policy	3
1.2.3 Replacement Policy	3
1.2.3.1. Local Fixed Partition Policies	4
1.2.3.2. Global Policies	4
1.2.3.3. Local Variable Partition Policies	5
1.3. Performance Considerations	5
1.3.1. The Locality Principle	5
1.3.2. Sources of Performance Degradation	6
1.3.3. Evaluating Virtual Memory Performance	6
1.3.4. Local Indices of Virtual Memory Performance	7
1.3.5. Two-Dimensional Representations	8
1.3.6. The Space-Time Product	9
1.3.7. Virtual Memory Tuning	11
1.4. Concluding Remarks	12
2. The Restructuring Approach	13
2.1. Introduction	13
2.2. Organization of Program Restructuring Procedures	13
2.3. Analysis of the Clustering Phase	14
2.3.1. Defining Affinities between more than Two Blocks	14
2.3.2. Algorithm Complexity Issues	15
2.3.3. Conclusion	15
2.4. Analysis of the Restructuring Phase	16
2.4.1. Static Algorithms	16
2.4.2. Dynamic Algorithms	16
2.4.3. Review of Dynamic Algorithms	17
2.5. Strategy-Oriented Restructuring	18
2.5.1. Critical Algorithms	18
2.5.2. Minimal Algorithms	19