Acid-Free Copy

SURF: AN ABSTRACT MODEL OF DISTRIBUTED GARBAGE COLLECTION

William Brodie-Tyrrell

February 2008

A THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE SCHOOL OF COMPUTER SCIENCE
UNIVERSITY OF ADELAIDE

© Copyright 2008
by
William Brodie-Tyrrell

Contents

Abstract **v**

Declaration **vii**

Acknowledgments **ix**

1 Introduction **1**

1.1 Garbage Collection 3
1.2 Distribution 6
1.3 Distributed Object Stores and GC 8
1.4 Models of (Distributed) Garbage Collection 12
1.5 Contributions & Structure of Thesis 14

2 Distributed Storage Management **17**

2.1 Distributed Computation 18
2.2 Distributed Termination Detection 19
2.3 Distributed Garbage Collection 21
2.4 Models of Distributed Garbage Collection 31
2.5 Requirement for a New Model of Distributed GC 38
2.6 Summary 38

3 Unifying Distributed Garbage Collection **41**

3.1 System Model 42
3.2 Definition of Garbage Collection 45
3.3 Surf: the Abstract Model of GC 48
3.4 Proving Safety and Completeness of Surf 61
3.5 Instantiating the Model 74
3.6 Limitations of the Model 82
3.7 Conclusion 83

4 Applying the Surf Model **85**
4.1 Distributed Marking 86
4.2 Train Algorithm 90
4.3 Hughes' Algorithm 94
4.4 Back Tracing 97
4.5 Reference Counting 103
4.6 Trial Deletion 105
4.7 Conclusion 111

5 Experimenting With Trains **113**
5.1 Mapping 114
5.2 Design 119
5.3 Experimentation 134
5.4 Conclusion 150

6 The Tram Algorithm **153**
6.1 Design of the Tram Algorithm 154
6.2 Discovering Topology 165
6.3 Comparisons with Other Collectors 169
6.4 Conclusions 175

7 Conclusion **177**
7.1 Overview 177
7.2 Further Work 180
7.3 Conclusion 181

Bibliography **183**

List of Figures

1 Reachability Examples 5
2 Relativistic Light Cone 7
3 Causality in Distributed Systems 8
4 Erroneous Reference Count 24
5 Erroneous Reference Count 24
6 (Usefully) Dead Regions 50
7 Inter-Region Pointers are Work 52
8 Distance Heuristic for Suspicion 102
9 DPMOS Architecture 119
10 Progress by Younger-First 131
11 Progress by Older-First 131
12 Layered Architecture for Measurement 136
13 Mesh of Triangles 139
14 Grid of Meshes 139
15 Complexity to Completion, FEA 144
16 Cost to Completion, FEA 144
17 Remembered Set Cache Performance 145
18 Page Cache Performance 146
19 Accuracy of Progress Prediction, OO7 147
20 Accuracy of Progress Prediction, FEA 149
21 Progress Histogram, Reverse FIFO 149
22 Witness Request Protocol 158
23 Witness Request Denied 158
24 Labelling from Heuristic 163
25 Optimally Labelled Regions 173

Abstract

Garbage collectors (GCs) automate the problem of deciding when objects are no longer reachable and therefore should be reclaimed, however, there currently exists no automated process for the design of a correct garbage collector. Formal models exist that prove the correctness of individual GCs; more general models describe a wider range of GCs but do not prove their correctness or provide a concrete instantiation process. The lack of a formal model means that GCs have been designed in an ad-hoc manner, published without proof of correctness and with bugs; it also means that it is difficult to apply experience gained from one implementation to the design of another.

This thesis presents **Surf**, an abstract model of distributed garbage collection that bridges the gap between expressibility and specificity: it can describe a wide range of GCs and contains a proof of correctness that defines a list of requirements that must be fulfilled. Surf's design space and its requirements for correctness provide a process that may be followed to analyse an existing collector or create a new GC.

Surf predicts the abstract behaviour of GCs; this thesis evaluates those predictions in light of the understood behaviour of published GCs to confirm the accuracy of the model. A distributed persistent implementation of the Train Algorithm is created as an instantiation of Surf and the model is used to analyse progress in the GC and drive the design of a partition selection policy that provides a lower bound on progress and therefore reduces the GC's complexity to completeness. Tests with mesh data structures from finite element analysis confirm the progress predictions from Surf.

Published GCs cluster mostly in one corner of the Surf design space so this thesis explores the design of a GC at an unoccupied design point: the Tram Algorithm. Analysis via Surf leads to the prediction that Trams are capable of discovering topology in the live object graph that approximately identifies the strongly connected components, permitting $O(1)$ timeliness that is unique to the Tram Algorithm.

Declaration

This work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text.

I give consent to this copy of my thesis, when deposited in the University of Adelaide Library, being made available in all forms of media, now or hereafter known.

William Brodie-Tyrrell
May 3, 2008

Acknowledgments

I would like to thank my supervisors, Assoc. Prof. David S. Munro and Dr Katrina Falkner, for their unceasing support, insight, guidance and motivation; this thesis would not exist without their Herculean efforts. The Jacaranda Research Group has provided a friendly research environment; I thank Dr Henry Detmold for the breadth and sharpness of his insight, the postgraduate students of the School for their time, encouragement and camaraderie, and the staff of the School for their support.

I also thank my family for their years of unquestioning support and encouragement. In particular, my parents have demonstrated that the term "standing on the shoulders of giants" refers not only to the work of previous academics.

This work was supported by the Commonwealth of Australia through an Australian Postgraduate Award and Australian Research Scholarship (National). Cluster computation time for experimentation was donated by the South Australian Partnership for Advanced Computing.