

A New Approach to The Train Algorithm For Distributed Garbage Collection

Matthew Clifton Lowry

December 2004

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

Contents

Abstract	viii
Declaration	ix
Acknowledgements	x
1. Introduction	1
1.1. The Garbage Collection Problem	3
1.1.1. Qualities of Garbage Collection Solutions	5
1.2. The Distributed Garbage Collection Problem	8
1.2.1. Qualities of Distributed Garbage Collection Solutions	10
1.3. Contribution and Overview	14
2. Solutions To The Garbage Collection Problem	17
2.1. Isolation Detection	17
2.1.1. Single-Object Isolation Detection	18
2.1.2. Subgraph Isolation Detection	18
2.2. Contained Segregation	21
2.2.1. Single-Site Systems	22
2.2.2. Distributed Systems	26
2.3. Hybridisation for Scalability with Completeness	29
2.4. Constructive Frameworks	31
2.5. Interaction with Other Distributed Object System Technologies	34
2.6. Models of Garbage Collection Behaviour	35
2.7. Chapter Summary	38
3. The Train Algorithm	40
3.1. Abstract Train Algorithm Model	40
3.1.1. System State Form	41
3.1.2. Mutator Event Forms	43
3.1.3. Collector Event Forms	45
3.1.4. Guaranteeing Complete Collection	48
3.1.5. Model Review	53
3.2. Asynchronous Metadata Updates	57
3.2.1. Refinements	57
3.2.2. Safety and Completeness	60

3.3.	Building Trains From Cars	64
3.3.1.	Refinements	65
3.3.2.	Safety and Completeness	69
3.4.	Chapter Summary	71
4.	Mechanisms For Distributed Train Collection	73
4.1.	Distributed Object System Requirements	73
4.2.	Distributed Train Management	77
4.3.	Local Collection and Reassociation	82
4.4.	Distributed Pointer Tracking	87
4.5.	Distributed Isolated Train Detection	94
4.5.1.	Adaption Of Distributed Termination Detection Algorithms	94
4.5.2.	Closure and Wave-Based Detection	97
4.5.3.	Pointer Record Ignorance	103
4.5.4.	Example Derivations	109
4.6.	Behavioural Properties	116
4.6.1.	Safety	116
4.6.2.	Completeness	125
4.6.3.	Scalability	127
4.7.	Chapter Summary	130
4.7.1.	Comparison to Prior Distributed Train Approaches	132
5.	Implementing Distributed Train Collection	136
5.1.	Overview of the CANDI Simulation System	137
5.1.1.	Simulation Engine Design	141
5.2.	Implementing Distributed Train Collection in CANDI	143
5.2.1.	Concurrent Access to Collector Metadata	143
5.2.2.	Limits to Concurrency Between Mutation and Collection	145
5.2.3.	Train Management Implementation	148
5.3.	Observing Collection Behaviour in CANDI Simulations	149
5.4.	Chapter Summary	159
6.	Conclusion	160
6.1.	Suggestions for Additional Research	161

A. Train Algorithm Model Summary	165
A.1. Basic Model	165
A.2. Asynchronous Metadata Updates	168
A.3. Building Trains From Cars	171
B. Collector Optimisations	174
B.1. Local Collection Policies	174
B.2. Reducing Pointer Tracking Overhead	176
B.2.1. Cancelling and Compressing Pointer Records	177
B.2.2. Reducing Root Reference Tracking Overhead	178
B.2.3. Reducing Reassociation Overhead	180
References	183

List of Figures

1.	Graph-of-objects model of program state.	4
2.	Accessibility equated to reachability.	5
3.	Distributed object graph	9
4.	Public and private garbage objects and scalability	13
5.	Race between collector and mutator activity	50
6.	Graphical depiction of the abstract Train Algorithm model	53
7.	Abstract Train Algorithm example	54
8.	Graphical depiction of the refined Train Algorithm model	71
9.	Car partitioning and refsets example	74
10.	Remote references operated via indirection	77
11.	Train identifier ordering	78
12.	Pointer record transmission	91
13.	Deriving isolation detection mechanisms from DTD algorithms	95
14.	Allocation causing spontaneous car activity	98
15.	Reassociation causing spontaneous car activity	98
16.	Ring-based wave topology	109
17.	Ring-based termination detection	110
18.	Ring-based isolation detection mechanism	111
19.	Tree-based wave topology	112
20.	Tree-based termination detection (slave aspect)	114
21.	Tree-based isolation detection mechanism (slave aspect)	115
22.	CANDI system overview	137
23.	CANDI Virtual Site details	138
24.	Distributed object system behaviour	140
25.	Race between collector and mutator over pointer record delivery	146
26.	Specialising local collector behaviour with root and nursery trains	175
27.	Conglomerating pointer records after cancelling or compression	178
28.	Avoiding pointer record generation for volatile references	179

List of Tables

1.	Contained segregation categories in single-site systems	22
2.	Contained segregation categories in distributed systems	26
3.	High-quality distributed algorithms via hybridisation	30
4.	Train isolation mapped to the DTD paradigm	96
5.	Car and train behaviour corresponding to DTD assumptions	97
6.	Object demographics at end of execution; SimpleTest workload	154
7.	Private object demographics at end of execution; Monte Workload	157
8.	Public object demographics at end of execution; Monte Workload	158
9.	State components of the abstract Train Algorithm model	165
10.	Mutator events of the abstract Train Algorithm model	166
11.	Collector events of the abstract Train Algorithm model	166
12.	Constraints of the abstract Train Algorithm model	167
13.	Mutator events for the asynchronous metadata update refinement	168
14.	Collector events for the asynchronous metadata update refinement	169
15.	Constraints for the asynchronous metadata update refinement	170
16.	State components of the car-based refinement	171
17.	Mutator events of the car-based refinement	172
18.	Collector events of the car-based refinement	173
19.	Constraints of the car-based refinement	173

Abstract

This thesis describes a new approach to achieving high quality distributed garbage collection using the Train Algorithm. This algorithm has been investigated for its ability to provide high quality collection in a variety of contexts, including persistent object systems and distributed object systems. Prior literature on the distributed Train Algorithm suggests that safe, complete, asynchronous, and scalable collection can be attained, however an approach that achieves this combination of behaviour has yet to emerge. The mechanisms and policies described in this thesis are unique in their ability to exploit the distributed Train Algorithm in a manner that displays all four desirable qualities. Further the mechanisms allow any number of mutator and collector threads to operate concurrently within a site; this is also a unique property amongst train-based mechanisms (distributed or otherwise).

Confidence in the quality of the approach promoted in this thesis is obtained via a top-down approach. Firstly a concise behavioural model is introduced to capture fundamental requirements for safe and complete behaviour from train-based collection mechanisms. The model abstracts over the techniques previously introduced under the banner of the Train Algorithm. It serves as a self-contained template for correct train-based collection that is independent of a target object system for deployment of the algorithm. Secondly a means to instantiate the model in a distributed object system is described. The instantiation includes well-established techniques from prior literature, and via the model these are correctly refined and reorganised with new techniques to achieve asynchrony, scalability, and support for concurrency. The result is a flexible approach that allows a distributed system to exhibit a variety of local collection mechanisms and policies, while ensuring their interaction is safe, complete, asynchronous, and scalable regardless of the local choices made by each site.

Additional confidence in the properties of the new approach is obtained from implementation within a distributed object system simulation. The implementation provides some insight into the practical issues that arise through the combination of distribution, concurrent execution within sites, and train-based collection. Executions of the simulation system are used to verify that safe collection is observed at all times, and obtain evidence that asynchrony, scalability, and concurrency can be observed in practice.

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 consent to this copy of my thesis, when deposited in the University Library, being available for loan and photocopying.

Matthew Clifton Lowry,
15 December 2004.

Acknowledgements

Firstly I would like to thank my supervisor Dr David Munro for his guidance and whip-cracking through the years in which the research presented here was undertaken. I am also grateful to my colleagues in the Jacaranda Research Group for refusing to believe my blithe assertions and arguing with me. In particular I am very grateful to Dr Katrina Falkner for effectively being an unofficial second supervisor. I must also give many thanks to my family and friends for the unlimited support they provided and their incessant credulity that I would eventually complete this work.

Also I would like to acknowledge the contributions (mostly inadvertent, I suspect) that have been made by the many researchers who have or continue to investigate the Train Algorithm; our interaction through workshops and research visits has certainly influenced the outcome of my own efforts. Lastly I would like to acknowledge the financial support I've received through the Australian Postgraduate Association, The University of Adelaide, the Australian Research Council, and my parents for making all this possible.

Typeset in Concrete and Euler using L^AT_EX.