

**ARMS: A DECENTRALISED NAMING MODEL FOR
OBJECT-BASED DISTRIBUTED COMPUTING SYSTEMS**

By

Jiabin Li, B.Eng. (Hons) (Curtin)

THIS THESIS IS PRESENTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY OF
MURDOCH UNIVERSITY

2010

DECLARATION

I declare that this thesis is my own account of my research and contains as its main content work that has not previously been submitted for a degree at any tertiary education institution.

Jiabin Li

ACKNOWLEDGEMENTS

I would like to begin by expressing my immense gratitude and appreciate to my supervisors Associate Professor Dr. Lance Chun Che Fung and Dr. Douglas G. Myers for their advice, support, guidance and inspiration throughout the course of my study. Professor Fung deserves special thanks for all of his help, especially with the preparation of this thesis, his comments and interest in my work from when I was an Honours student until now. He is not only my mentor, but also a most valued friend whom I am forever grateful to have.

Many thanks to Associate Professor Dr. Kevin Kok Wai Wong and Dr. Kien-Ping Chung for their assistance during my time at Murdoch University. Thanks also to all the staff in the School of Information Technology, Murdoch University.

Finally, without the support of my parents and my wife, their understanding and patience, it would have been impossible for me to complete this study.

ABSTRACT

Entities communicate with one another in distributed computing systems via symbolic names. Implementing such communication requires a naming scheme that dynamically maps these symbolic names to physical nodes and processes. Traditionally, a centralised name server is deployed to perform such translations. However, a collaborative and dynamic environment requires a decentralised naming system due to reasons of efficiency and reliability.

ARMS (Adaptive, Randomised and Migration-enabled Scheme) is a novel decentralised naming scheme for distributed object-oriented computing systems. A notable feature of ARMS is that it provides direct naming supports for the patterns of object communication and object migration processes to achieve greater performance and scalability in executing object-oriented software within a distributed environment. These supports are driven by three key components: 1) *an adaptive locating protocol* that exploits the patterns of object communication and explores the best routing path in the face of the changing network conditions, 2) *a randomised overlay* that is a scalable and flexible substrate for routing name queries, and 3) *a hybrid relocation scheme* that provides a transparent and efficient means of referencing migrated objects.

The performance of ARMS has been examined using a number of real world Java-based benchmarking programs. Based on results in this study, ARMS has found to be superior to its structural counterpart – the *Chord model* because of the adaptive routing protocol and the resilient overlay. Furthermore, ARMS has shown to be superior in a number of other performance metrics.

TABLE OF CONTENTS

DECLARATION.....	I
ACKNOWLEDGEMENTS.....	II
ABSTRACT	III
TABLE OF CONTENTS.....	IV
LIST OF FIGURES	VIII
LIST OF TABLES	XV
LIST OF PUBLICATIONS.....	XVII
1. INTRODUCTION.....	1
1.1. NAMING ISSUES IN OBJECT-BASED DISTRIBUTED COMPUTING SYSTEMS	1
1.2. ARMS: AN EFFICIENT DECENTRALISED NAMING MODEL.....	2
1.3. THESIS CONTRIBUTIONS	4
1.4. THESIS STRUCTURE	6
2. A STUDY OF DYNAMIC NAMING SYSTEMS	8
2.1. OVERVIEW	8
2.2. NAME CONVENTIONS	8
2.3. BINDING AND RESOLUTION	10
2.4. HIERARCHICAL LOOKUP SYSTEMS	14
2.5. DECENTRALISED NAMING SYSTEMS.....	21
2.6. CLASSIFICATION OF PEER-BASED NAMING SYSTEMS.....	26
2.6.1. <i>Ad hoc routing on unstructured overlay (ARUO)</i>	26
2.6.2. <i>Static routing on structured overlay (SRSO)</i>	31
2.6.3. <i>Informed routing on constrained overlay (IRCO)</i>	41

2.6.4. <i>Proximate routing on randomised overlay (PRRO)</i>	47
3. A NOVEL DECENTRALISED NAMING SCHEME DIRECTED AT OBJECT-BASED COMPUTING STRUCTURES	53
3.1. NAMING SUPPORTS FOR MESSAGE PASSING AND OBJECT MIGRATION	53
3.2. PLACEMENT OF A RANDOM GRAPH OVER THE CHORD TOPOLOGY.....	56
3.3. ADAPTIVE LOCATING PROTOCOL	59
3.3.1. <i>Ant-inspired routing</i>	59
3.3.2. <i>Algorithm description</i>	60
3.3.3. <i>Updating mechanism</i>	65
3.3.4. <i>Proof of correctness</i>	68
3.4. PERFORMANCE IMPROVEMENT VIA CONSTRUCTIVE HEURISTIC INFORMATION.....	70
3.5. TRANSPARENT OBJECT RELOCATION SCHEME	71
3.6. A COMPARISON OF ARMS AND OTHER PEER-BASED NAMING SYSTEMS.....	77
4. ANALYSIS OF PATTERNS IN OBJECT COMMUNICATION	80
4.1. SELECTION OF JAVA-BASED BENCHMARK PROGRAMS	80
4.2. CHARACTERISTICS OF OBJECTS AND THEIR MESSAGE PASSING.....	82
4.2.1. <i>Object dynamics</i>	82
4.2.2. <i>Non-uniform object communication</i>	87
4.2.3. <i>Exploring locality in object communication</i>	87
4.3. DISCUSSION ON PROFILING RESULTS	90
5. SIMULATION STUDY OF THE PERFORMANCE CHARACTERISTICS OF THE ADAPTIVE LOCATING PROTOCOL.....	92
5.1. PERFORMANCE MEASURES.....	92
5.2. INVESTIGATION OF PARAMETER SETTINGS.....	94
5.2.1. <i>Test description</i>	94
5.2.2. <i>Pheromone importance factor (PIF)</i>	95

5.2.3. <i>Heuristic importance factor (HIF)</i>	100
5.2.4. <i>Evaporation rate (ER)</i>	101
5.2.5. <i>Decay rate (DR)</i>	103
5.2.6. <i>Heuristic constant (HC)</i>	104
5.2.7. <i>Pheromone upper bound (PUB)</i>	106
5.2.8. <i>A summary of the test results</i>	107
5.3. A STUDY OF LOOKUP SCALABILITY	109
5.3.1. <i>State of forwarding table</i>	109
5.3.2. <i>Size of parallel query messages</i>	111
5.3.3. <i>Network size</i>	113
5.3.4. <i>Object population</i>	116
5.4. IMPLEMENTATION ON DIFFERENT NETWORK STRUCTURES.....	119
6. IMPACTS OF CONSTRUCTIVE HEURISTIC INFORMATION ON THE LOCATING EFFICIENCY	122
6.1. PATH EXPLORATION FOR IMPROVING OBJECT LOCATING PERFORMANCE.....	122
6.2. OPTIMISATION STRATEGIES.....	124
6.2.1. <i>Optimising for proximity</i>	124
6.2.2. <i>Load balancing optimisation</i>	129
6.2.3. <i>Degree of connectivity</i>	133
6.2.4. <i>Optimisation based on spatial locality</i>	136
6.3. PERFORMANCE ANALYSIS	139
7. A HYBRID SCHEME FOR LOCATING MIGRATED OBJECTS.....	148
7.1. OVERVIEW OF RELOCATION SCHEMES	148
7.2. PERFORMANCE MEASURES.....	151
7.3. SIMULATION ANALYSIS.....	153
7.3.1. <i>Scaling with the migration counts</i>	153

7.3.2. <i>Impact of messaging intervals</i>	157
7.3.3. <i>Scaling with the network sizes</i>	157
7.4. DISCUSSION ON SIMULATION RESULTS	160
8. CONCLUSIONS	162
8.1. EXTENSIVE SUPPORTS FOR RUNTIME OPTIMISATION TO IMPROVE ROUTING EFFICIENCY	162
8.2. SEAMLESS INTEGRATION OF THE NAMING PROCESS AND THE ROUTING PROCESS	164
8.3. PROVISION OF A SCALABLE TRANSPARENT RELOCATION SCHEME FOR REFERENCING MIGRATED OBJECTS	165
8.4. REMARKS ON THE JAVA-BASED BENCHMARK PROGRAMS.....	166
8.5. IMPROVEMENTS TO THE SYSTEM	167
APPENDIX: A DISCRETE EVENT, TRACE DRIVEN SIMULATOR FOR STUDYING OBJECT-BASED DISTRIBUTED COMPUTING SYSTEMS	170
BIBLIOGRAPHY	175

LIST OF FIGURES

FIGURE 1: AN ILLUSTRATION OF THE THREE FORMS OF MESSAGE ROUTING: (A) ROUTING BASED ON LOCATION DEPENDENT NAMES; (B) ROUTING WITH LOCATION INDEPENDENT NAMES VIA A NAME SERVER; (C) ROUTING DIRECTLY BASED ON LOCATION INDEPENDENT NAMES.	5
FIGURE 2: AN ILLUSTRATION OF THREE AD-HOC SEARCH ALGORITHMS: FLOODING, PROBABILISTIC FLOODING AND N RANDOM WALKERS ($N=2$).....	31
FIGURE 3: THE TRAVELLED PATH ON A TAPESTRY NETWORK BY A QUERY MESSAGE WITH THE REQUESTED KEY <i>1742</i> . THE QUERY WAS INITIATED AT PEER <i>294C</i> AND DESTINED AT PEER <i>1742</i>	35
FIGURE 4: AN EXAMPLE OF A 3-BIT CHORD RING WITH 6 ACTIVE PEERS AS PEERS 3 AND 6 DO NOT EXIST. A QUERY PATH WAS ESTABLISHED FOR THE LOOKUP OF KEY <i>0</i> ORIGINATED AT PEER <i>1</i>	37
FIGURE 5: A SUMMARY OF PEER-TO-PEER (P2P) OVERLAY NETWORKS.	52
FIGURE 6: AN ILLUSTRATION OF THE CHORD RING ARCHITECTURE.	57
FIGURE 7: THE OBJECT LOCATION PROTOCOL DEFINED IN ARMS.	63
FIGURE 8: AN ALGORITHM THAT UPDATES PHEROMONES VIA BACKWARD PROPAGATION. A REPLY MESSAGE TRAVERSES IN THE REVERSE OF THE QUERYING PATH.	67
FIGURE 9: THE STATE DIAGRAM OF THE OBJECT LOCATING PROTOCOL.	70
FIGURE 10: THE TROMBONE PROBLEM WHERE THE DISTANCE OF LINKS <i>A</i> , <i>B</i> IS MUCH GREATER THAN LINK <i>C</i>	72
FIGURE 11: THE REVISED UPDATE PROTOCOL. IT UPDATES BOTH THE PHEROMONES OF A SEARCH PATH AND THE FORWARD CHAIN.	74
FIGURE 12: THE OBJECT RELOCATION PROTOCOL USED IN ARMS.	74
FIGURE 13: THE STATE DIAGRAM OF ARMS RELOCATION SCHEME.	75

FIGURE 14: THE FLOW OF THE METHODOLOGY ADOPTED IN THIS STUDY.....	82
FIGURE 15: A DEMONSTRATION OF THE PERCENTAGE OF THE SAMPLING OBJECT POPULATION TO THE OVERALL POPULATION FOR FOUR BENCHMARK PROGRAMS AS THE PROGRESS OF THE EXECUTION.	83
FIGURE 16: AN ILLUSTRATION OF THE NUMBER OF OBJECT CREATIONS [ABOVE] AND POTENTIAL OBJECT DELETIONS [BELOW] AS THE FUNCTION OF THE NORMALISED EXECUTION TIME IN THE AUTOFOCUS.....	85
FIGURE 17: AN ILLUSTRATION OF THE NUMBER OF OBJECT CREATIONS [ABOVE] AND POTENTIAL OBJECT DELETIONS [BELOW] AS THE FUNCTION OF THE NORMALISED EXECUTION TIME IN THE DYNAMICJAVA.....	85
FIGURE 18: AN ILLUSTRATION OF THE NUMBER OF OBJECT CREATIONS [ABOVE] AND POTENTIAL OBJECT DELETIONS [BELOW] AS THE FUNCTION OF THE NORMALISED EXECUTION TIME IN THE IMAGEJ.....	86
FIGURE 19: AN ILLUSTRATION OF THE NUMBER OF OBJECT CREATIONS [ABOVE] AND POTENTIAL OBJECT DELETIONS [BELOW] AS THE FUNCTION OF THE NORMALISED EXECUTION TIME IN THE RHINO.....	86
FIGURE 20: AN EXAMPLE OF SOURCE CODE OF THE IJ CLASS CONTAINED IN THE BENCHMARK IMAGEJ.	89
FIGURE 21: THE PHEROMONE IMPORTANT FACTOR VERSUS THE SEARCH PATH FOR FOUR BENCHMARKS.....	97
FIGURE 22: THE PHEROMONE IMPORTANCE FACTOR VERSUS THE SIMULATION COMPLETION TIME FOR FOUR BENCHMARKS.....	97
FIGURE 23: THE PATH LENGTH VERSUS THE HEURISTIC IMPORTANCE FACTOR FOR FOUR BENCHMARKS.....	100
FIGURE 24: THE HEURISTIC IMPORTANCE FACTOR VERSUS THE SIMULATION COMPLETION TIME FOR FOUR BENCHMARKS.	101

FIGURE 25: THE EVAPORATION RATE VERSUS THE LOOKUP PATH LENGTH FOR FOUR BENCHMARKS.....	103
FIGURE 26: THE EVAPORATION RATE VERSUS THE SIMULATION COMPLETION TIME.....	104
FIGURE 27: THE HEURISTIC CONSTANT VERSUS THE AVERAGE LOCATING PATH LENGTH.	105
FIGURE 28: THE HEURISTIC CONSTANT VERSUS THE SIMULATION COMPLETION TIME. ...	105
FIGURE 29: THE PHEROMONE UPPER BOUND VERSUS THE AVERAGE LOCATING LENGTH.	106
FIGURE 30: THE PHEROMONE UPPER BOUND VERSUS THE SIMULATION COMPLETION TIME.	107
FIGURE 31: A COMPARISON OF THE ROUTING STATE BETWEEN ARMS AND THE CHORD FOR EACH BENCHMARK.	110
FIGURE 32: AN ILLUSTRATION OF THE PROGRESSING ROUTING STATE IN ARMS FOR EACH BENCHMARK.	110
FIGURE 33: A DEMONSTRATION OF THE LOOKUP LENGTH AS THE FUNCTION OF THE AMOUNT OF THE INITIAL QUERY MESSAGES. THE INITIAL MESSAGES WERE CHANGING FROM 10% TO 100% OF THE SIZE OF CANDIDATE NEIGHBOURS.	112
FIGURE 34: A DEMONSTRATION OF THE EXECUTION TIME AS THE FUNCTION OF THE AMOUNT OF INITIAL QUERY MESSAGES THE INITIAL MESSAGES WERE CHANGING FROM 10% TO 100% OF THE SIZE OF CANDIDATE NEIGHBOURS.	112
FIGURE 35: A LOGARITHMIC PLOT OF THE LENGTH OF THE LOOKUP PATH AS THE FUNCTION OF THE NETWORK SIZE FOR DYNAMICJAVA. EACH COLUMN SHOWS THREE VALUES FROM THE TOP TO THE BOTTOM: THE LONGEST LOOKUP PATH, THE EXPECTED LOOKUP PATH AND THE SHORTEST LOOKUP PATH AT EACH NETWORK SIZE.	114
FIGURE 36: A LOGARITHMIC PLOT OF THE LENGTH OF THE LOOKUP LENGTH AS THE FUNCTION OF THE NETWORK SIZE FOR IMAGEJ. EACH COLUMN SHOWS THREE VALUES	

FROM THE TOP TO THE BOTTOM: THE LONGEST LOOKUP PATH, THE EXPECTED LOOKUP PATH AND THE SHORTEST LOOKUP PATH AT EACH NETWORK SIZE.....	114
FIGURE 37: A PLOT OF THE LENGTH OF THE EXECUTION TIME AS THE FUNCTION OF THE NETWORK SIZE FOR DYNAMICJAVA.....	115
FIGURE 38: A PLOT OF THE LENGTH OF THE EXECUTION TIME AS THE FUNCTION OF THE NETWORK SIZE FOR IMAGEJ.	115
FIGURE 39: A PLOT OF THE LOOKUP LENGTH AS SIMULATING DIFFERENT SIZES OF THE OBJECT POPULATIONS USING THE BENCHMARK IMAGEJ.	116
FIGURE 40: A PLOT OF THE LOOKUP LENGTH AS SIMULATING DIFFERENT SIZES OF THE OBJECT POPULATIONS USING THE BENCHMARK RHINO.....	117
FIGURE 41: A PLOT OF THE EXECUTION TIME AS SIMULATING DIFFERENT SIZES OF THE OBJECT POPULATIONS USING THE BENCHMARK IMAGEJ.	117
FIGURE 42: A PLOT OF THE EXECUTION TIME AS SIMULATING DIFFERENT SIZES OF THE OBJECT POPULATIONS USING THE BENCHMARK RHINO.....	118
FIGURE 43: A PLOT OF THE EXECUTION TIME AS SIMULATING DIFFERENT SIZES OF THE OBJECT POPULATIONS GENERATED BY A RANDOM MODEL.....	118
FIGURE 44: AN ILLUSTRATION OF THE LOOKUP LENGTH AS SIMULATING ARMS ON TWO TYPES OF NETWORK STRUCTURES, 2D GRID AND CUBE.	120
FIGURE 45: AN ILLUSTRATION OF THE EXECUTION TIME AS SIMULATING ARMS ON TWO TYPES OF NETWORK STRUCTURES, 2D GRID AND CUBE.	121
FIGURE 46: AN ILLUSTRATION OF THE VALUE OF THE MEAN LOOKUP LENGTH (MLL) FOR THE ARM_LD, THE ARM_PR AND THE CHORD SIMULATED WITH FOUR BENCHMARK PROGRAMS.	126
FIGURE 47: AN ILLUSTRATION OF THE VALUE OF THE DISTANCE PER QUERY (DPQ) FOR THE ARM_LD, THE ARM_PR AND THE CHORD SIMULATED WITH FOUR BENCHMARK PROGRAMS.	126

FIGURE 48: AN ILLUSTRATION OF THE VALUE OF THE WORKLOAD STANDARD DEVIATION (WSD) FOR THE ARM_LD, THE ARM_LB AND THE CHORD THROUGH SIMULATING THE BENCHMARK RHINO.	130
FIGURE 49: AN ILLUSTRATION OF THE VALUES OF THE MEAN LOOKUP LENGTH (MLL) FOR THE ARM_LD, THE ARM_LB AND THE CHORD SIMULATED WITH FOUR BENCHMARK PROGRAMS.	132
FIGURE 50: AN ILLUSTRATION OF THE VALUE OF THE AVERAGE WORKLOAD (AWL) FOR THE ARM_LD, THE ARM_LB AND THE CHORD SIMULATED WITH FOUR BENCHMARK PROGRAMS.	132
FIGURE 51: AN ILLUSTRATION OF THE EXECUTION TIME FOR ALL BENCHMARK PROGRAMS WITH SIMULATING THE ARM_LD AND THE ARM_DOC ON THE NETWORK OF 2704 NODES.	135
FIGURE 52: AN ILLUSTRATION OF THE AVERAGE WORKLOAD FOR ALL BENCHMARK PROGRAMS WITH SIMULATING THE ARM_LD AND THE ARM_DOC ON THE NETWORK OF 2704 NODES.	135
FIGURE 53: AN ILLUSTRATION OF THE STANDARD DEVIATION FOR WORKLOADS FOR ALL BENCHMARK PROGRAMS WITH SIMULATING THE ARM_LD AND THE ARM_DOC ON THE NETWORK OF 2704 NODES.	136
FIGURE 54: AN ILLUSTRATION OF THE MEAN LOOKUP LENGTH FOR THE ARM_LD AND THE ARM_SL SIMULATED WITH FOUR BENCHMARK PROGRAMS ON THE NETWORK OF 1156 NODES.	137
FIGURE 55: AN ILLUSTRATION OF THE MEAN LOOKUP LENGTH FOR ALL SCHEMES AS THE FUNCTION OF THE NETWORK SIZE THROUGH SIMULATING THE BENCHMARK AUTOFOCUS.	140

FIGURE 56: AN ILLUSTRATION OF THE MEAN LOOKUP LENGTH FOR ALL SCHEMES AS THE FUNCTION OF THE NETWORK SIZE THROUGH SIMULATING THE BENCHMARK DYNAMICJAVA.....	140
FIGURE 57: AN ILLUSTRATION OF THE MEAN LOOKUP LENGTH FOR ALL SCHEMES AS THE FUNCTION OF THE NETWORK SIZE THROUGH SIMULATING THE BENCHMARK IMAGEJ.	141
FIGURE 58: AN ILLUSTRATION OF THE MEAN LOOKUP LENGTH FOR ALL SCHEMES AS THE FUNCTION OF THE NETWORK SIZE THROUGH SIMULATING THE BENCHMARK RHINO.	141
FIGURE 59: AN ILLUSTRATION OF THE EXECUTION TIME FOR ALL SCHEMES AS THE FUNCTION OF THE NETWORK SIZE BASED ON THE BENCHMARK AUTOFOCUS.	143
FIGURE 60: AN ILLUSTRATION OF THE EXECUTION TIME FOR ALL SCHEMES AS THE FUNCTION OF THE NETWORK SIZE BASED ON THE BENCHMARK DYNAMICJAVA.	143
FIGURE 61: AN ILLUSTRATION OF THE EXECUTION TIME FOR ALL SCHEMES AS THE FUNCTION OF THE NETWORK SIZE BASED ON THE BENCHMARK IMAGEJ.	144
FIGURE 62: AN ILLUSTRATION OF THE EXECUTION TIME FOR ALL SCHEMES AS THE FUNCTION OF THE NETWORK SIZE BASED ON THE BENCHMARK RHINO.....	145
FIGURE 63: AN ILLUSTRATION OF THE MEAN LOOKUP LENGTH FOR ALL SCHEMES AS SIMULATED ON A NETWORK OF 2116 NODES.	146
FIGURE 64: AN ILLUSTRATION OF THE DISTANCE PER QUERY (DPQ) FOR ALL SCHEMES AS SIMULATED ON A NETWORK OF 2116 NODES.	146
FIGURE 65: AN ILLUSTRATION OF THE STANDARD DEVIATION FOR WORKLOADS (WSD) FOR ALL SCHEMES AS SIMULATED ON A NETWORK OF 2116 NODES.	147
FIGURE 66: AN ILLUSTRATION OF THE THREE COMMON RELOCATION SCHEMES: (A) HOME- BASED; (B) FORWARD POINTERS; (C) BACKTRACKING.	149
FIGURE 67: AN ILLUSTRATION OF THE AVERAGE MIGRATION TIME OF THE THREE SCHEMES WITH RESPECT TO THE MIGRATION COUNTS.....	155

FIGURE 68: DISTRIBUTION OF THE QUERY LOADS RECEIVED BY A HOME PEER IN THE FUNCTION OF THE MIGRATION COUNTS FOR EACH RELOCATION SCHEME.....	155
FIGURE 69: AN ILLUSTRATION OF THE QUERY PATH LENGTH OF THE THREE RELOCATION SCHEMES IN THE FUNCTION OF MIGRATION COUNT.....	156
FIGURE 70: AN ILLUSTRATION OF THE AVERAGE INVOCATION TIME FOR THE THREE RELOCATION SCHEMES, AS FUNCTION OF THE MIGRATION COUNTS.	156
FIGURE 71: A DEMONSTRATION OF THE AVERAGE MIGRATION TIME AT VARIOUS MESSAGING RATES.	158
FIGURE 72: A DEMONSTRATION OF THE AVERAGE INVOCATION TIME AT VARIOUS MESSAGING RATES.	159
FIGURE 73: THE QUERY PATH LENGTH VERSES THE NETWORK SIZES UNDER THE DIFFERENT NETWORK SIZES.....	159
FIGURE 74: THE MIGRATION TIME VERSES NETWORK SIZES UNDER THE DIFFERENT NETWORK SIZES.....	160
FIGURE 75: AN ABSTRACT REPRESENTATION OF A COMPUTING NODE DEFINED IN THE DISCRETE EVENT SIMULATOR.	170
FIGURE 76: A UML (UNIFIED MODELLING LANGUAGE) CLASS DIAGRAM OF THE DISCRETE EVENT SIMULATOR.	173
FIGURE 77: AN ACTIVITY DIAGRAM OF THE DISCRETE EVENT SIMULATOR.	174

LIST OF TABLES

TABLE 1: A COMPARISON OF VARIOUS STRUCTURED PEER-TO-PEER NETWORKS.	34
TABLE 2: A SNAPSHOT OF THE FORWARDING TABLE AT NODE N3 IN ARMS.	59
TABLE 3: AN ILLUSTRATION OF THE DISTRIBUTION OF THE LIFESPAN OF JAVA OBJECTS. THE LIFE SPAN IS MEASURED AS THE PERCENTAGE OF THE TOTAL EXECUTION TIME.	84
TABLE 4: A DEMONSTRATION OF THE FREQUENCY OF INVOCATIONS SENT OR RECEIVED BY OBJECTS IN EACH BENCHMARK PROGRAM. THE LAST ROW SHOWS THE CORRELATION OF THE NUMBER OF SENT MESSAGES AND THE NUMBER OF RECEIVED MESSAGES BY THE SAME OBJECT IN A BENCHMARK.	88
TABLE 5: AN ILLUSTRATION OF THE PROFILING RESULTS FOR THE TOTAL NUMBER OF OBJECT CALLS, THE NUMBER OF THE CLASS CALLS, THE NUMBER OF INHERITANCE CALLS, THE NUMBER OF TEMPORAL CALLS.	88
TABLE 6: IMPACT OF THE DIFFERENT CONFIGURATIONS ON THE LENGTH OF THE LOOKUP PATH FOR FOUR BENCHMARK PROGRAMS. THE SHORTEST PATH LENGTH IN EACH CATEGORY HAS BEEN HIGHLIGHTED.	98
TABLE 7: IMPACT OF THE DIFFERENT CONFIGURATIONS ON THE EXECUTION TIME FOR FOUR BENCHMARK PROGRAMS. THE LOWEST EXECUTION TIME IN EACH CATEGORY HAS BEEN HIGHLIGHTED.	99
TABLE 8: A SUMMARY OF THE TEST RESULTS FOR THE TOTAL AMOUNT OF QUERY MESSAGES, THE TOTAL DISTANCE TRAVELLING BY THE MESSAGES, AND THE AVERAGE TRAVELLING DISTANCE PER MESSAGE.	121
TABLE 9: LISTING OF THE TEST RESULTS FROM SIMULATING ARMS WITH THE HEURISTIC INFORMATION DESIGNED FOR THE LOOKUP LENGTH UNDER VARIOUS NETWORK SIZES.	127

TABLE 10: LISTING OF THE RESULTS FROM SIMULATING ARMS WITH THE HEURISTIC INFORMATION DESIGNED FOR THE PROXIMITY UNDER VARIOUS NETWORK SIZES. . .	128
TABLE 11: A DEMONSTRATION OF THE TEST RESULTS FOR SIMULATING ARMS WITH THE HEURISTIC INFORMATION DESIGNED FOR LOAD BALANCING UNDER VARIOUS NETWORK SIZES.....	131
TABLE 12: LISTING OF TEST RESULTS FOR SIMULATING ARMS WITH THE HEURISTIC INFORMATION DESIGNED FOR THE DEGREE OF CONNECTIVITY UNDER VARIOUS NETWORK SIZES.....	134
TABLE 13: LISTING OF THE PROFILING RESULTS FOR FOUR BENCHMARK PROGRAMS REGARDING THE TOTAL REMOTE CALLS, THE SPATIAL CALLS, THE CLASS CALLS AND THE INHERITANCE CALLS.....	138
TABLE 14: LISTING OF THE TEST RESULTS FOR SIMULATING ARMS WITH THE HEURISTIC INFORMATION DESIGNED FOR THE SPATIAL LOCALITY IN VARIOUS NETWORK SIZES.	138

LIST OF PUBLICATIONS

The following publications (in reverse chronicle order) were produced in the course of the research leading to this thesis. A taxonomy for classifying decentralised naming systems was first proposed in 2004 [9]. Subsequently, the focus of the work was on the design of a discrete-event simulation framework in 2005 [7,8] and a Java-based profiler in 2006 [6]. Results from the study of object locating mechanisms in an object-based distributed computing system were later reported in 2007 and 2008 [2-5]. Finally, the research of the transparent relocation scheme was published in 2010 [1].

- [1] “*Migration Supports for Potential Mobile Objects in Ubiquitous Computing Systems*”, Proceedings of the International Conference of Impact on Ubiquitous IT Co-Design to Industry, Perth, Australia, 21–23, January 2010.
- [2] “*Application of Ant Colony Optimisation in Object Locating in A Distributed Computing System*”, Proceedings of the Second IEEE International Conference on Digital Ecosystem and Technologies (IEEE-DEST 2008), 26-29 February, 2008, Phitsanulok, Thailand, ISBN: 1-4244-1490-3, pp 59-64.
- [3] “*An Adaptive Randomised Structured Search Network for Locating Objects in a Distributed Computing System*”, Proceedings of the Eighth Postgraduate Electrical Engineering and Computing Symposium (PEECS 2007), Curtin University of Technology, Western Australia, November 2007, ISBN 1-74067-5673, pp. 17-22.
- [4] “*Intelligent Object Locality Naming Model in An Object-based Distributed System for Engineering Applications*”, Proceedings of the Sixth International Conference on Machine Learning and Cybernetics 2007 (ICMLC’07), Hong Kong, China, 19-22 August 2007, ISBN 1-4244-0972-1, pp 124-130.

- [5] “*Efficient Locating and Relocating Scheme for Object-based Distributed Systems*”, Proceedings of the 2007 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA’07), Las Vegas, Nevada, USA, 25-28 June, 2007, ISBN 1-60132-020-5, pp 69-75.
- [6] “*Characteristics of Java Class File for Code Optimisation in a Distributed Computing Environment*”, Proceedings of the 7th Postgraduate Electrical and Computing Symposium (PEECS 2006), ISBN 86905-977-7, Murdoch University, Perth, WA, pp. 13-17.
- [7] “*Evaluation of a Small Scale Cluster Computing System for Parallel Intelligent Techniques Applications*”, Proceedings of the Fourth International Conference on Machine Learning and Cybernetics (ICMLC 2005), ISBN 0-7803-9092-X, 19 - 21 August 2005, Guangzhou, China, pp 388-393.
- [8] “*A Simulation Design for Evaluating Naming Models in Object-based Distributed Systems*”, Proceedings of the 6th Postgraduate Electrical and Computing Symposium (PEECS ’05), ISBN 0-7298-06090-X, Edith Cowan University, Perth, WA, pp. 166-171.
- [9] “*Naming Models in Object-based Distributed Systems*”, Proceedings of the *Fifth* Postgraduate Electrical Engineering and Computing Symposium (PEECS 2004), September 2004, Perth, Western Australia, pp 124-129.