Contents

Acknowledgements vii

Contents xi

List of Figures xv

List of Tables xvii

Publications xix

1 General Introduction 1
- 1.1 Many-Task Computing . . . 2
- 1.2 MTC Application Storage Requirements . . . 3
- 1.3 In-Memory Computing . . . 3
- 1.4 Related Approaches . . . 5
- 1.5 Thesis Contributions and Outline . . . 6
  - 1.5.1 Chapter 2: MemFS - The No-Locality Approach . . . 7
  - 1.5.2 Chapter 3: MemEFS - Elasticity Improves Resource Utilization 7
  - 1.5.3 Chapter 4: MemEFS - Network-Awareness . . . 8
  - 1.5.4 Chapter 5: MemFSS - Memory Scavenging . . . 8

2 Overcoming Data Locality: an In-Memory Runtime File System with Symmetrical Data Distribution 11
- 2.1 Introduction . . . 12
- 2.2 Background and Related Work . . . 13
- 2.3 System Design . . . 16
  - 2.3.1 Components . . . 16

2.3.2 Implementation Building Blocks 17
2.4 Evaluation 20
2.4.1 MTC Envelope Evaluation 21
2.4.2 Application Benchmarks & Multicore Scalability 25
2.4.3 Summary & Discussion 39
2.5 Conclusions and Future Work 40

3 MemEFS: an Elastic In-Memory Runtime File System for eScience Applications 43
3.1 Introduction 44
3.2 Background 45
3.2.1 Elasticity Requirements 46
3.3 MemEFS Design and Implementation 47
3.3.1 Two-Layer Hashing Scheme 48
3.3.2 Load Balancing 49
3.3.3 Initialization and Reconfiguration 50
3.3.4 Implementation 51
3.4 Evaluation 53
3.4.1 Experimental Setup 53
3.4.2 Preliminary Results 54
3.4.3 Elastic Scaling Policies 56
3.4.4 MemEFS Performability 57
3.4.5 Storage Load Balancing 64
3.4.6 Discussion 65
3.5 Related Work 66
3.5.1 Elasticity in Data Storage Systems 66
3.5.2 Load Balancing Schemes in DHTs 67
3.6 Conclusion and Future Work 68

4 MemEFS - a Network-Aware In-Memory Distributed File System 69
4.1 Introduction 70
4.2 MemEFS Network Adaptation 70
4.2.1 MemEFS 71
4.2.2 Network Awareness 71
4.3 Network Adaptability Evaluation 73
4.3.1 Cloud Experimental Setup 73
4.3.2 Network Adaptability Experiments 74
4.3.3 Discussion 76
4.4 Related Work 79
4.5 Conclusions 79

5 Towards Resource Disaggregation - Memory Scavenging for Scientific Workloads 81
5.1 Introduction . . . 82
5.2 Motivation and Background . . . 83
5.2.1 The Achieved Parallelism of Scientific Workflows . . . 83
5.2.2 Cluster Resource Utilization . . . 84
5.2.3 MemFS . . . 85
5.3 MemFSS Design . . . 85
5.3.1 The MemFSS System . . . 85
5.3.2 Hashing . . . 86
5.3.3 POSIX (FUSE) Interface . . . 87
5.3.4 Data & Metadata Placement . . . 87
5.3.5 Fault-Tolerance . . . 88
5.3.6 Data Privacy & Isolation . . . 88
5.4 Evaluation . . . 89
5.4.1 Experimental Setup . . . 89
5.4.2 Scavenging Overhead Baseline . . . 90
5.4.3 Real-World Application Slowdown . . . 93
5.4.4 Resource Consumption Reduction . . . 97
5.4.5 Summary . . . 97
5.5 Related Work . . . 98
5.5.1 Cycle Scavenging - Volunteer Computing . . . 98
5.5.2 Remote Memory Usage . . . 99
5.5.3 Weighted Hashing . . . 99
5.6 Conclusions . . . 100

6 General Conclusions 101
6.1 Thesis Contributions . . . 102
6.2 Future Directions . . . 103

References 105

Summary 117

Samenvatting 121