Table of Contents

Chapter One - Introduction 1

1.1. Basic Concepts 4

1.2. Application-Specific Techniques 7

1.3. Application-Transparent Techniques 9

1.3.1. Message Logging 9

1.3.2. Coordinated Checkpointing 11

1.4. Scope of the Thesis 14

1.5. Thesis Organization 18

Chapter Two - System Model 24

2.1. Computational Model 24

2.1.1. Fault-Free Operation 25

2.1.2. Operation with Failures and Recovery 26

2.2. Assumptions 29

2.3. Interprocess Communication 30

2.4. Error Detection 31

Chapter Three - Message Logging Techniques 35

3.1. Previous Work 36

3.2. Data Structures for Asynchronous Message Logging 40

3.3. Basic Approach of Message Logging 41

iv

3.4. Recovering a Consistent Global State 45

3.5. Algorithms 50

3.5.1. Normal Operation and Checkpointing 50

3.5.2. Recovery 52

3.6. Proofs of Correctness 55

3.6.1. Deleting Regenerated Messages 56

3.6.2. Handling Orphan Messages 58

3.6.3. Messages Following Ordered Playback 62

3.6.4. Summary 68

3.7. Other Issues 69

3.8. Summary 71

Chapter Four - Coordinated Checkpointing Techniques 73

4.1. Previous Work 76

4.2. Basic Approach of Coordinated Checkpointing 79

4.2.1. Recording a Consistent Global State 79

4.2.2. Global Checkpointing 82

4.2.3. Recording Interacting Sets of Processes 83

4.2.4. Maintaining a Consistent Global State 83

4.2.5. Recovery From Consistent Global States 90

4.3. Coordinating Interacting Sets as Trees 93

4.3.1. Coordinating the Checkpoint/Recovery Sessions 95

4.3.1.1. Multiple Independent Checkpoint/Recovery Sessions 98

4.3.1.2. Multiple Coordinators 105
4.4. Synchronous Coordinated Checkpointing 114
4.4.1. Supporting Dynamic Virtual Channels and Processes 115
4.4.2. Synchronous Checkpointing of Interacting Sets 118
4.4.3. Concurrent Invocations of the Algorithms 121
4.4.4. Checkpointing with Packet-Switching 122
4.5. Recovery 123
4.5.1. Recovery From Communication Errors 123
4.5.2. Recovery From Node Failures 124
4.5.3. Handling Failures during Checkpoint Sessions 125
4.6. Asynchronous Coordinated Checkpointing 128
4.6.1. Asynchronous Checkpointing: 1 Coordinator, 1 Session 133
4.6.2. Asynchronous Checkpointing: N Coordinators, 1 Session 135
4.6.3. Asynchronous Checkpointing: N Coordinators, N Sessions 136
4.6.4. Summary 139
4.7. Minimizing the Cost of Checkpointing Using the Virtual Memory System 140
4.8. Conclusions 142
Chapter Five - Execution-Driven Simulation of Error Recovery Techniques 144
5.1. Introduction 145

vi

5.2. Testbed Design Considerations 147

5.3. The Testbed 148

5.3.1. The Interconnection Network 150

5.3.2. Simulating the Execution of Application Processes 152

5.3.3. The Operating System 155

5.3.4. The Host Processor (I/O Object) and Stable Storage (Disk Object) 156

5.3.5. Summary 157

5.4. The Recovery Schemes Simulated 157

5.5. Experiences with the Applications 160

5.6. Experiences with the Recovery Schemes 164

5.6.1. Message Logging 166

5.6.2. Coordinated Checkpointing 169

5.6.3. Optimizations 171

5.7. Simulator Performance 173

5.8. Previous Work 174

5.9. Summary and Conclusions 175

Chapter Six - Performance Evaluation 177

6.1. Detailed Itemization of Recovery Scheme Overhead 177

6.1.1. Overhead Of Checkpointing/Recovery of Independent Processes 178

6.1.2. Overhead of Asynchronous Message Logging 184

6.1.2.1. The Cost of Logging Messages to Disk 186

vii

6.1.2.2. Other Message Logging Techniques 187

6.1.3. Overhead Of Coordinated Checkpointing of Interacting Sets 194

6.2. Performance Evaluation Using DERT 198

6.2.1. Performance of Asynchronous Message Logging 200

6.2.1.1. Overhead Due to Network Bandwidth 209

6.2.1.2. Overhead Due to Disk Bandwidth 211

6.2.1.3. Reducing the Number of Orphan Processes 213

6.2.1.4. Committing Outputs to the External World 216

6.2.2. Performance of Coordinated Checkpointing 218

6.2.2.1. Overhead Due to Network Bandwidth 227

6.2.2.2. Overhead Due to Disk Bandwidth 229

6.2.2.3. Simulations with Increased Checkpoint Frequency 230

Chapter Seven - Conclusions 236

Bibliography 242

viii