# A MODULAR AND EXTENSIBLE NETWORK STORAGE ARCHITECTURE

**Distinguished Dissertations in Computer Science**

Edited by
C.J. van Rijsbergen, University of Glasgow

The Conference of Professors of Computer Science (CPCS), in conjunction with the British Computer Society (BCS), selects annually for publication up to four of the best British PhD dissertations in computer science. The scheme began in 1990. Its aim is to make more visible the significant contribution made by Britain – in particular by students – to computer science, and to provide a model for future students. Dissertations are selected on behalf of CPCS by a panel whose members are:

C.B. Jones, Manchester University (Chairman)
S. Abramsky, Imperial College, London
H.G. Barrow, University of Sussex
D.A. Duce, Rutherford Appleton Laboratory
M.E. Dyer, University of Leeds
D. May, Inmos Ltd, Bristol
V.J. Rayward-Smith, University of East Anglia
M.H. Williams, Heriot-Watt University

# A MODULAR AND EXTENSIBLE NETWORK STORAGE ARCHITECTURE

SAI-LAI LO
*Downing College Cambridge*
*Research Engineer, Olivetti Research Limited, Cambridge*

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town,
Singapore, São Paulo, Delhi, Tokyo, Mexico City

Cambridge University Press
The Edinburgh Building, Cambridge CB2 8RU, UK

Published in the United States of America by Cambridge University Press, New York

www.cambridge.org
Information on this title: www.cambridge.org/9780521551151



First published 1995

*A catalogue record for this publication is available from the British Library*

ISBN 978-0-521-55115-1Hardback

To my parents

# Contents

**List of Figures** xii

**List of Tables** xiii

**Glossary** xv

**Preface to this edition** xvii

**Preface** xix

**1 Introduction** 1
- 1.1 What is a network storage service? . . . 1
- 1.2 Research Motivation . . . 2
- 1.3 Research Statement . . . 2
- 1.4 Dissertation Outline . . . 2

**2 Background** 5
- 2.1 Introduction . . . 5
- 2.2 The Development of Distributed File Systems . . . 5
- 2.3 Limitations of Today's Distributed File Systems . . . 7
- 2.4 Support of Continuous-Medium Data . . . 8
- 2.5 Support of Structured Data . . . 10
- 2.6 Extensions to the Primary Storage Function . . . 11
  - 2.6.1 File Indexing . . . 11
  - 2.6.2 Persistent Programming Languages . . . 12
- 2.7 Better Data Placement Strategies . . . 13
- 2.8 Summary . . . 15

**3 Architectural Framework** 17
3.1 Goals . . . 17
3.2 MSSA Entities . . . 17
3.3 Storage Layers . . . 19
3.3.1 Rationale . . . 20
3.3.2 PS Layer . . . 21
3.3.3 LS Layer . . . 22
3.4 Custodes . . . 23
3.5 Containers . . . 24
3.6 Mapping . . . 26
3.7 The Byte Segment Abstraction . . . 27
3.7.1 Data Abstraction . . . 27
3.7.2 Operations . . . 27
3.8 BSC Sessions and Tickets . . . 29
3.8.1 Rationale . . . 29
3.8.2 Session Interactions . . . 29
3.9 Related Work . . . 30
3.9.1 Modular File System Design . . . 31
3.9.2 HLSS and LLSS . . . 31
3.9.3 DataMesh . . . 32
3.10 Summary . . . 33

**4 Access control** 35
4.1 Protection Requirements . . . 35
4.2 Authorisation . . . 37
4.3 Authentication . . . 38
4.4 Access control in MSSA . . . 39
4.5 The Use of Access Control Lists in MSSA . . . 39
4.6 The Use of Capabilities in MSSA . . . 41
4.7 Summary . . . 47

**5 Naming and Related Issues** 49
5.1 Textual Names vs Identifiers . . . 49
5.2 Naming . . . 51
5.2.1 Considerations . . . 51
5.2.2 Container and Object Identifiers . . . 52
5.2.3 Generating Object Identifiers . . . 53
5.2.4 Locating Containers and Objects . . . 55
5.2.5 Naming and Value-Adding Clients . . . 55
5.3 Existence Control . . . 56
5.4 Summary . . . 57

**6 The Design of a Byte Segment Custode** 59
6.1 Introduction . . . 59
6.2 Design Considerations . . . 59
6.2.1 Failure Recovery . . . 59
6.2.2 Failure Recovery in MSSA . . . 60

6.2.3 NVRAM and Atomic Updates . . . 60
6.3 NVRAM Transactions . . . 62
6.3.1 Overview . . . 62
6.3.2 NVRAM Buffer Blocks . . . 63
6.3.3 Intention Lists . . . 65
6.3.4 Committing Transactions . . . 66
6.3.5 Recovery . . . 66
6.3.6 A Loose End . . . 67
6.4 Metadata . . . 67
6.5 Disk Block Allocation . . . 74
6.6 Buffering and Disk I/O . . . 74
6.7 Other Implementation Details . . . 76
6.8 Summary . . . 77

**7 The Performance of the BSC 79**
7.1 Performance . . . 79
7.1.1 Best-case Performance . . . 79
7.1.2 Performance Cost of Atomic Writes . . . 80
7.1.3 Recovery Time . . . 83
7.1.4 I/O Throughput . . . 83
7.2 Related Work . . . 85
7.2.1 Existing systems . . . 86
7.2.2 Performance Studies . . . 86
7.2.3 LLSS . . . 87
7.3 Summary . . . 88

**8 Rate-Based Sessions: Concept & Interface 89**
8.1 Introduction . . . 89
8.2 The CFC and the Translator . . . 90
8.2.1 The CFC . . . 90
8.2.2 The Translator . . . 91
8.3 Resource Reservation and Scheduling . . . 93
8.3.1 The Need to Reserve Resources . . . 93
8.3.2 Rate-based Sessions . . . 94
8.3.3 The Difficulties in Resource Reservation . . . 94
8.3.4 The Semantics of Rate-based Sessions . . . 95
8.4 Rate-Based Sliding Window . . . 95
8.4.1 Definition . . . 96
8.4.2 Relation with read-ahead scheduling . . . 96
8.4.3 Relation with byte segment accesses . . . 98
8.4.4 Variable rate sessions . . . 98
8.5 Rate-Based Session Interface . . . 98
8.5.1 Session Set-up and Shut-down . . . 99
8.5.2 Dynamic Window Adjustments . . . 100
8.6 Summary . . . 102

**9 Rate-based Sessions: Prototype Implementation & Evaluation** **103**
9.1 Prototype Implementation . . . 103
9.1.1 Progress Monitoring . . . 105
9.1.2 Storage Allocation . . . 105
9.1.3 Read-ahead and Write-behind Scheduling . . . 105
9.2 Evaluation . . . 107
9.2.1 Measured Parameters . . . 107
9.2.2 Experimental Setup . . . 108
9.2.3 Single Session Performance . . . 109
9.2.4 Multiple Session Performance . . . 112

**10 Conclusion** **117**
10.1 Summary . . . 117
10.2 Further Work . . . 119

**Bibliography** **120**

**Index** **131**

# List of Figures

2.1 Sample frame-size-profiles of MPEG . . . 9
2.2 Automatic Attribute Extraction in a Semantic File System . . . 12
2.3 MSSA Value-Adding Clients . . . 13
2.4 Location Dependency of Distributed File Systems . . . 14

3.1 MSSA Entities . . . 18
3.2 MSSA Layers . . . 19
3.3 MSSA Internal Structure . . . 20
3.4 MSSA Container Mapping . . . 25
3.5 Generalised File System Model . . . 31

4.1 Access Paths . . . 36
4.2 Capability format . . . 42
4.3 Making Capabilities Unforgeable . . . 43
4.4 Using the comment field . . . 46

5.1 The formats of container and object identifiers . . . 53

6.1 The Organisation of the NVRAM Buffer . . . 64
6.2 Valid State Transitions of a NVRAM Buffer Block . . . 64
6.3 The Intention List of a Transaction . . . 65
6.4 On-disk Metadata Structure . . . 68
6.5 A Sample Byte Segment Extent List . . . 70
6.6 On-Disk Representation of an Extent List . . . 71
6.7 Inserting Entries in the Middle of an Extent List . . . 73

7.1 Cost Breakdown of Atomic Write . . . 81
7.2 Cost of Atomic Write . . . 82
7.3 I/O throughput . . . 84

8.1 The CFC interface . . . 90

8.2 Handling Stream Heterogeneity . . . . . 92
8.3 Sliding Window & Read-Ahead Scheduling . . . . . 97
8.4 Sliding Window Adjustment . . . . . 100
8.5 Variable-rate Session and BSC Read-Ahead . . . . . 101

9.1 Rate-based Session Implementation . . . . . 104
9.2 Cumulative Distribution of Single Session Data Access Service Time . . . 109
9.3 Cumulative Distribution of Single Session Read-ahead Service Time . . . . 111
9.4 Cumulative Distribution of Multiple Session Data Access Service Time . . . 113
9.5 Cumulative Distribution of Multiple Session Read-ahead Service Time . . . 115

# List of Tables

6.1 NVRAM Buffer Block I/O State Table . . . . . 75

7.1 The Best-Case Performance of the BSC . . . . . 80

9.1 Average Read-ahead Service Time with Multiple Sessions . . . . . 113

## Trademarks

UNIX is a registered trademark of AT&T.
NFS is a trademark of Sun Microsystems, Inc.

# Glossary

This list defines abbreviations and some technical terms used in the text.

**ATM** Asynchronous Transfer Mode.

**BSC** Byte Segment Custode. A physical storage layer custode that implements the byte segment abstraction.

**byte segment** A physical storage layer object which abstracts the secondary storage device.

**CFC** Continuous-media File Custode. A logical storage layer custode that implements the continuous-media file abstraction.

**CSCAN** Circular SCAN. A disk scheduling algorithm.

**continuous-media** Data types, such as digital video and audio, which are sequences of discrete and temporally related data samples.

**continuous-media file** A file type to store continuous-media data.

**custode** n. One who has the custody of anything- Oxford English Dictionary 1971. A server of MSSA.

**DCE** Distributed Computing Environment. A network of client workstations with shared resources provided by a group of servers.

**FFC** Flat File Custode. A logical storage layer custode that implements the flat file abstraction.

**flat file** A synonym of byte-stream file. This corresponds to the file abstraction of conventional filing system such as UNIX.

**logical storage layer** The upper layer of MSSA and consists of different file custodes. Each custode implements a file abstraction.

**LS layer** See logical storage layer.

**MSSA** The Multi-Service Storage Architecture. A network storage service design investigated in this dissertation.

**NVRAM** Non-volatile random access memory.

**physical storage layer** The lower layer of MSSA and consists of byte segment custodes.

**PS layer** See physical storage layer.

**SATF** Shortest access time first. A disk scheduling algorithm.

**SFC** Structured File Custode. A LS layer custode that implements the structured file abstraction.

**structured file** A file type of MSSA which can store arbitrary user-defined structures.

**UPS** Uninterruptible power supply

**Value-adding client** A client of MSSA that provides a service to other clients.

# Preface to this edition

This edition is largely an un-edited version of my dissertation that was submitted over a year ago. An index is added to help readers to locate the sections of interest to them. The work that is described in this book is part of a research project[1] that I continued to work on for another year after I had written my dissertation. In a way this book represents a snapshot of an ongoing project that continues to evolve, as any active research work should be.

Some of the ideas described here have been further developed and refined. More flexible use of access control lists has been introduced. The identifiers used in the system are no longer structured, hence the name space is more efficiently used and value-adding clients are better supported. Stream interleaving is introduced to allow the continuous-medium custode to deliver streams with a dynamic range of quality of service. Over the past year or so, my colleagues and I at the Computer Laboratory, University of Cambridge, have completed the implementation of most components described in this book.

However I have resisted the temptation to update the text, partly to keep in line with the spirit of this book series, and partly because I believe the design principles, which are the main contributions of my work, have not been changed. I shall leave it to our publications more recently to report the progress of this project.

I wish to thank the selection committee, chaired by Professor van Rijsbergen of the University of Glasgow, for the honour of receiving this distinguished dissertation award and the Cambridge University Press for publishing this work.

[1] The project is funded by the UK EPSRC grant GR/H 13666.

# Preface

I would like to thank my superviser, Jean Bacon, for her advice, encouragement, and practical assistance during the course of this research.

I am grateful to Ken Moody and members of the System Research Group for their valuable advice and helpful discussions. Tim Wilson, Sue Thomson, Glenford Mapp and Richard Hayton deserve special mention. I am also grateful to the past and present members of the Laboratory who develop and maintain the WANDA system. This target platform is instrumental to the experimental work that was done in this research. I appreciate the assistance provided by Jean Bacon, Ken Moody, Richard Hayton, Glenford Mapp, Robert Sultana, Tim Wilson and John Bates who suggested improvements to the dissertation.

I am deeply grateful to my family for their love and support. My parents' unceasing emphasis on education and assiduousness has provided me with many opportunities.

I am indebted to the finiancial support provided by the Croucher Foundation and the Science and Engineering Research Council.

Except where otherwise stated in the text, this dissertation is the result of my own work and is not the outcome of work done in collaboration.

I hereby declare that this dissertation is not substantially the same as any I have submitted for a degree or diploma or any other qualification at any other university.

I further state that no part of my dissertation has already been, or is being currently submitted for any such degree, diploma or other qualification.