

# **Breaking the Availability Barrier III**

*Active/Active Systems in Practice*

**Paul J. Holenstein  
Dr. Bruce Holenstein  
Dr. Bill Highleyman**

*AuthorHouse™*  
*1663 Liberty Drive, Suite 200*  
*Bloomington, IN 47403*  
*www.authorhouse.com*  
*Phone: 1-800-839-8640*

© 2007 Paul J. Holenstein, Dr. Bruce Holenstein, and Dr. Bill Highleyman. All rights reserved.

*No part of this book may be reproduced, stored in a retrieval system, or transmitted by any means without the written permission of the authors.*

*First published by AuthorHouse 5/24/2007*

*ISBN: 978-1-4343-1606-6 (sc)*  
*ISBN: 978-1-4343-1607-3 (hc)*  
*ISBN: 978-14343-1608-0 (e)*

*Printed in the United States of America*  
*Bloomington, Indiana*

*This book is printed on acid-free paper.*

*All products mentioned in this book are trademarks of their respective owners.*  
*The information in this book is provided on an as-is informational basis.*  
*The authors, owners, and publisher disclaim liability for any errors or omissions.*  
*The reader accepts all risks associated with the use of the contents of this book.*



## **Dedication**

**This book is dedicated to our spouses,  
Karen, Denise, and Janice,  
for their enduring patience and support.**

**We also dedicate this book to Jim Gray  
for his fundamental contributions to transaction  
processing technology on which this book is based.  
Jim, an avid sailor, has been missing at sea  
since January 28, 2007.**



# *Breaking the Availability Barrier III*

## **Contents**

<b>Forward .....</b>	<b>i</b>
What is “This Book”? .....	ii
Achieving Extreme Availabilities .....	iii
A Roadmap Through “This Book” .....	vii
Part 1 – Survivable Systems for Enterprise Computing .....	viii
Part 2 – Building and Managing Active/Active Systems .....	ix
Part 3 – Infrastructure Case Study .....	xi
Part 4 – Active/Active Systems at Work .....	xii
Appendices .....	xiii
Authors’ Notes .....	xiv
Acknowledgements.....	xv
About the Authors .....	xvi
<b>Part 3 – Infrastructure Case Study.....</b>	<b>1</b>
<b>Chapter 10 – Performance of Active/Active Systems .....</b>	<b>3</b>
Overview.....	4
Model Assumptions .....	5
Replication Engine Architecture.....	7
Replicator Performance Examples .....	11
Performance Comparison .....	15
Sources of Latency .....	17
Throughput Bottlenecks.....	18
Application Latency.....	19
Coordinated Commits.....	19
Network Transactions.....	20
Communication Channel Efficiency.....	20
Synchronous Replication Efficiency .....	21
An Example .....	23

## *Breaking the Availability Barrier III*

Effect On Response Time .....	25
What's Next.....	26
<b>Chapter 11 - Shadowbase .....</b>	<b>27</b>
The Ideal Replication Engine .....	27
The Shadowbase Data Replication Engine .....	29
Problems That Shadowbase Solves .....	29
Business Continuity .....	30
<i>What Do We Mean by Business Continuity?</i> .....	30
<i>Replicating Systems for High Availability</i> .....	30
<i>Optimizing RTO</i> .....	30
<i>Optimizing RPO</i> .....	33
<i>Zero Planned Downtime</i> .....	35
Data Collisions.....	35
Data Synchronization.....	36
<i>Query</i> .....	36
<i>Offloading Backups and Warehouse Extracts</i> .....	36
<i>Active/Active Systems</i> .....	38
Data Integration .....	38
<i>Operational Data Store (ODS)</i> .....	39
<i>Active Data Warehousing</i> .....	39
<i>Real-Time Business Intelligence</i> .....	40
Utility Uses .....	40
Shadowbase Principles .....	42
The Shadowbase Architecture .....	44
Architectural Overview.....	44
Supported Databases.....	46
<i>Source Systems</i> .....	46
<i>Target Systems</i> .....	48
Homogeneous Replication.....	48
<i>NonStop Servers</i> .....	49
<i>Oracle, SQL Server, and Sybase</i> .....	51
NonStop/Open Replication .....	53
Replicating to Other Systems.....	54
Data Transformation .....	55
<i>Transformation and Mapping Facility</i> .....	56

## *Breaking the Availability Barrier III*

<i>User Exits</i> .....	56
Synchronous Data Replication .....	56
Bidirectional Replication .....	57
Additional Options.....	59
<i>Shadowbase AutoLoader</i> .....	59
<i>Shadowbase Online Loading, Validation and Verification</i> (SOLV) .....	59
The Benefits of Shadowbase .....	60
What's Next? .....	62
<b>Chapter 12 - SOLV .....</b>	<b>63</b>
What is SOLV? .....	63
The Problems that SOLV Solves .....	63
Data Replication .....	64
<i>Backup for Disaster Recovery</i> .....	66
<i>Query</i> .....	66
<i>Active/Active Systems</i> .....	66
<i>Online Upgrades</i> .....	67
Online Loading and Copying.....	67
Real-Time Business Intelligence .....	68
Online Verification and Validation .....	70
Online Resynchronization .....	70
Hurdles to Online Copying .....	71
Offline Extraction, Transformation, and Loading (Offline ETL) ..	71
Online Extraction, Transformation and Loading (Online ETL) ..	72
<i>The Problem</i> .....	73
<i>Queuing Updates</i> .....	74
<i>Copying Via the Replication Stream</i> .....	76
The SOLV Solution .....	78
SOLV .....	78
<i>The SOLV Architecture</i> .....	79
<i>Multithreading SOLV for Performance</i> .....	80
<i>Verification and Validation</i> .....	81
<i>Resynchronization</i> .....	82
<i>Active/Active Configurations</i> .....	82
<i>Database Consistency</i> .....	83
<i>Fuzzy Copy</i> .....	84

## *Breaking the Availability Barrier III*

What's Next?.....85

### **Chapter 13 – ZDM with Shadowbase.....87**

Eliminating Planned Downtime.....87

    ZDM in Review ..... 88

    ZDM with Shadowbase and SOLV ..... 92

Other Uses for ZDM .....94

    Capacity Expansion ..... 94

    Load Balancing ..... 94

    Initial Migration to an Active/Active Environment..... 95

    Incremental Migration ..... 95

    Disaster Tolerance ..... 95

    Data Locality..... 95

    Lights-Out Operation..... 95

    Risk-Free Failover Testing ..... 95

What's Next.....96

### ***Part 4 – Active/Active Systems at Work..... 97***

#### **Chapter 14 – Benefits of Multiple Nodes in Practice ..... 99**

Extreme Availability.....100

Faster Recovery Time.....101

Elimination of Decision Time .....102

Disaster Tolerance for Free.....103

Improved Data Locality .....104

Eliminate Scheduled Downtime.....104

Efficient Use of All Capacity .....106

Risk-Free Failover Testing .....107

Application Scaling via Symmetric Expansion.....109

Application Scaling via Asymmetric Expansion .....109

Load Balancing .....112

Lights-Out Operation .....113

The Art of Compromise .....116

## *Breaking the Availability Barrier III*

What's Next ..... 117

### **Chapter 15 – Case Studies..... 119**

Is Active/Active Technology Real?..... 119

Homogeneous Active/Active Systems ..... 120

    Message Switching ..... 122

    Cell Phone Message Delivery..... 124

    Regional Bank Service Bureau..... 126

    OpenCall INS Cell Phone Application..... 128

    Prepaid Calling Card Voucher System..... 130

Heterogeneous Cooperative Processing Systems..... 131

    Credit Card Fraud Detection..... 132

    Railroad Fare Modeling..... 133

    Insurance Claims Processing..... 134

Disaster Recovery ..... 136

    Card Management Application..... 138

    Sizzling Hot Takeover..... 140

Asymmetric Capacity Expansion ..... 142

    Travel Agency Booking..... 144

    Plant Management ..... 146

    Real-Time Telephone Fraud Detection ..... 147

Zero Downtime Migrations ..... 149

    Casino Administration..... 151

    Paper Manufacturer ..... 152

    Master/Slave Cell Phone Application..... 153

What's Next? ..... 154

### **Chapter 16 – Related Technologies and Drivers 155**

The HP NonStop Advanced Architecture (NSAA)..... 155

    Early NonStop Systems ..... 155

    Drawbacks of Tightly-Coupled Lockstepping..... 158

    The NonStop Advanced Architecture (NSAA) ..... 159

    NSAA Failure Modes ..... 160

    DMR, TMR and SMR ..... 162

IBM's Parallel Sysplex ..... 164

## *Breaking the Availability Barrier III*

What is Parallel Sysplex?.....	164
Parallel Sysplex Architecture.....	164
Geographically Dispersed Parallel Sysplex.....	165
<b>The Data Integration Problem .....</b>	<b>167</b>
The Real-Time Enterprise (RTE).....	168
<i>What is RTE? .....</i>	<i>168</i>
<i>The Operational Data Store (ODS) .....</i>	<i>169</i>
Federated Databases.....	172
<i>A Unified View .....</i>	<i>172</i>
<i>Partitioned Federated Databases.....</i>	<i>175</i>
<i>Relation to Active/Active Technology .....</i>	<i>175</i>
<b>Split Mirrors .....</b>	<b>175</b>
The Problem of Lost Transactions.....	175
Split Mirrors.....	176
<b>Bulletproof Storage .....</b>	<b>179</b>
Bulletproof Storage in the Industry.....	179
IBM's Strategy.....	180
<b>Grid Computing .....</b>	<b>181</b>
What Is Grid Computing?.....	181
The History of Grid Computing.....	182
Grid Standards .....	183
Implementation .....	184
Uses for the Grid.....	185
Where Is the Grid Going?.....	185
<b>Virtual-Tape.....</b>	<b>186</b>
The Data Explosion.....	186
The Magnetic-Tape Conundrum.....	186
The Virtual-Tape Solution .....	188
<i>Reduced Cost.....</i>	<i>188</i>
<i>Improved Reliability .....</i>	<i>189</i>
<i>Faster Corrupted or Lost File Restoration .....</i>	<i>190</i>
<i>Faster System Recovery .....</i>	<i>190</i>
A Step Towards Higher Availability .....	191
<b>Regulatory Issues .....</b>	<b>192</b>
<b>What's Next.....</b>	<b>192</b>

# *Breaking the Availability Barrier III*

## **Appendices ..... 193**

### **Appendix 1 – Rules of Availability ..... 195**

Volume 1 Rules .....	195
Volume 2 Rules .....	199
Volume 3 Rules .....	202

### **Appendix 2 – Replication Engine Performance Model ..... 205**

The Replication Engine Model .....	205
Replication Transaction Flow .....	206
Replication Performance Model .....	208
Some Queuing Relationships .....	212
Processor Delay .....	213
Disk Delay .....	217
<i>Disk Queuing Delay</i> .....	217
<i>Disk Service Time</i> .....	218
Change Processing Time .....	219
Queue Poll Delay .....	221
Poll Pass .....	222
Interval .....	222
Arrive .....	222
Process .....	222
Wait .....	222
Communication Delay .....	224
<i>Buffering</i> .....	224
<i>Transmission</i> .....	225
<i>Propagation Time</i> .....	226
Commit Reserialization Delay .....	227
Parallelism .....	228
<i>Multithreaded Processes</i> .....	228
<i>Multiple Disk Volumes</i> .....	229
Filtering .....	230
Replication Performance Measures .....	231
Replication Latency .....	231

## *Breaking the Availability Barrier III*

Processor Loading.....	234
Disk Loading.....	234
Communication Line Loading.....	235
Throughput.....	235
<i>Load Limitations</i> .....	236
<i>Thread Limitations</i> .....	237
<i>Throughput</i> .....	238
Application Latency.....	238
Model Summary .....	239
Synchronous Replication Efficiency .....	249
Comparison of Application Latencies .....	249
Effect on Response Time .....	250
Commit Latency .....	252
Mean of an Exponential.....	257

### **Appendix 3 – Regulatory Requirements .....259**

BASEL II.....	259
B. C. Hydro Act .....	260
Centers for Medicare and Medicaid Services (CMS) .....	260
Control Objectives for Information and Related Technology (COBIT).....	260
Department of Energy (DOE).....	261
Environmental Protection Agency (EPA).....	261
EU Capital Requirements Directive (CRD).....	261
Federal Deposit Insurance Corp. (FDIC).....	262
Federal Drug Administration (FDA).....	262
FDA Title 21 Code of Federal Regulations Part 11 (21 CFR 11) .....	263
Federal Energy Regulatory Commission (FERC).....	263
Federal Financial Institutions Examination Council (FFIEC) .....	263
Federal Preparedness Circular (FPC) 65 .....	264
Federal Reserve Board (FRB).....	264
Financial Rating Agencies.....	264
Financial Services Modernization Act (also known as the Gramm-Leach-Bliley Act) .....	265

## *Breaking the Availability Barrier III*

Foreign Corrupt Practices Act (FCPA).....	265
Health Insurance Portability and Accountability Act (HIPAA)	265
Homeland Security Presidential Directive # 8 (HSPD-8) .....	266
Interagency Paper on Sound Practices to Strengthen the Resilience of the U.S. Financial System .....	266
International Accounting Standards Board (IASB).....	267
International Standards Organization (ISO).....	267
ISO 9000 .....	268
ISO 14000.....	268
ISO 15189.....	269
ISO 17799.....	269
Investment Dealers Association of Canada (IDA).....	269
Joint Commission on Accreditation of Healthcare Organizations (JCAHO) .....	270
KYC/AML/Patriot Act .....	270
Markets in Financial Instruments Directive (MiFID) .....	271
National Association of Security Dealers (NASD).....	271
NASD 3500.....	272
NASD 3510.....	272
NASD 3520.....	272
National Credit Union Administration (NCUA).....	273
National Environmental Research Council (NERC).....	273
National Fire Protection Association (NFPA 1600) .....	273
National Futures Association (NFA) Compliance Rule 2-38 .	274
National Institute of Standards and Technology (NIST) 800 .	274
New York Stock Exchange (NYSE) Rule 446.....	275
Occupational and Safety Health Act (OSHA).....	275
Office of the Comptroller of the Currency (OCC) .....	275
Office of Thrift Supervision (OTS).....	276
Patriot Act .....	276
Payment Card Industry (PCI) Data Security Standard .....	276
Sarbanes-Oxley Act (SOX).....	277
Scottish EPA Regulations (SEPA).....	277
Securities and Exchange Commission (SEC).....	277
SEC Title 17 Code of Federal Regulations (17 CFR) .....	278
Single European Payment Area (SEPA).....	278

## *Breaking the Availability Barrier III*

Solvency II.....	279
Three Pillars .....	279
<b>Appendix 4 – A Consultant’s Critique .....</b>	<b>281</b>
Authors’ Preface.....	281
Introduction and Overview.....	281
Why Multinode Architectures?.....	282
Scalability .....	282
Availability .....	282
Disaster Recovery .....	283
Types of Multinode Architectures .....	283
Shared Database.....	283
Partitioned Database .....	285
Replicated Database (Active/Active).....	288
Conclusion .....	290
Application Challenges with Multinode Architectures .....	290
General Considerations.....	290
Transaction Distribution .....	292
<i>General Remarks</i> .....	292
<i>Distribution Done by the Network</i> .....	292
<i>Distribution Done by the Application Front End</i> .....	293
Application Management.....	295
<i>Definitions</i> .....	295
<i>Application Configuration</i> .....	296
<i>Application Monitoring</i> .....	297
<i>Application Control</i> .....	298
<i>Conclusion</i> .....	299
Global Resources and Local Context.....	300
<i>Definitions</i> .....	300
<i>Example 1: Connection is Local Context</i> .....	300
<i>Example 2: A Unique Number Generator</i> .....	302
<i>Example 3: Application Decisions Based on Locking</i> .....	304
<i>Batch Considerations</i> .....	306
<i>Conclusion</i> .....	307
Response Time Behavior and Transaction Rates.....	307
Error Handling .....	309

## *Breaking the Availability Barrier III*

Final Remarks.....	310
<b>References and Suggested Reading .....</b>	<b>311</b>
<b>Index .....</b>	<b>319</b>
<b>About the Authors .....</b>	<b>341</b>







## Forward

*Given today's technology, [six 9s] is unachievable for all practical purposes, and an unrealistic goal.*

- Evan Marcus and Hal Stern, 2000<sup>1</sup>

My, how things change in just a few years! Not only are we going to talk about achieving systems with six 9s availability but also with eight 9s availability and beyond. Furthermore, we are not talking just about system availability. We are talking about application service availability. After all, following a failure of some sort, if the users of an application are being serviced in an unacceptable manner (such as experiencing excessively long response times), then the application is essentially *not available*.

If you could configure your current system to:

- provide extreme availability - MTBFs measured in centuries,
- affect only a subset of users upon a failure,
- recover from any failure in subseconds to seconds,
- lose little if any data as the result of a failure,
- eliminate planned downtime,
- achieve disaster tolerance,
- use all available capacity,
- load balance at will,
- be easily expandable,
- require no change to existing applications,
- all at little or no additional cost,

wouldn't you be interested? We think so, and that is what this book is all about. Active/active systems can and do provide these benefits today.

---

<sup>1</sup> Evan Marcus, Hal Stern, *Blueprints for High Availability: Designing Resilient Distributed Systems*, Wiley; 2000.

## *Breaking the Availability Barrier – Volume 3*

Abe Lincoln said that “it is better to remain silent and be thought a fool than to speak out and remove all doubt.” At the risk of sounding foolish to some, we recognize that there are naysayers who will argue that extreme availabilities cannot be achieved. In this book we are speaking out, confident that the many examples of successful installations of active/active systems will prove us not to be fools, notwithstanding Abe.

### **What is “This Book”?**

We referred to “this book” in the previous section. Actually, when we started to write “this book,” we intended it to be the second in a series on active/active systems. However, when we finished it, it became apparent that it was much too long to be a comfortable single book to read. Therefore, we decided to break it up into two volumes.

We will refer to the (now) three volumes as Volumes 1, 2, and 3. “This book” comprises Volumes 2 and 3. The titles of the active/active trilogy are:

Volume 1: *Breaking the Availability Barrier: Survivable Systems for Enterprise Computing*, published by AuthorHouse in 2004,

Volume 2: *Breaking the Availability Barrier II: Achieving Century Uptimes with Active/Active Systems*, published by AuthorHouse in 2007 along with Volume 3.

Volume 3: *Breaking the Availability Barrier III: Active/Active Systems in Practice*, published by AuthorHouse in 2007 along with Volume 2.

In keeping with Volumes 2 and 3 being essentially one book, this Forward is the same in each volume. However, the content of each volume is markedly different.

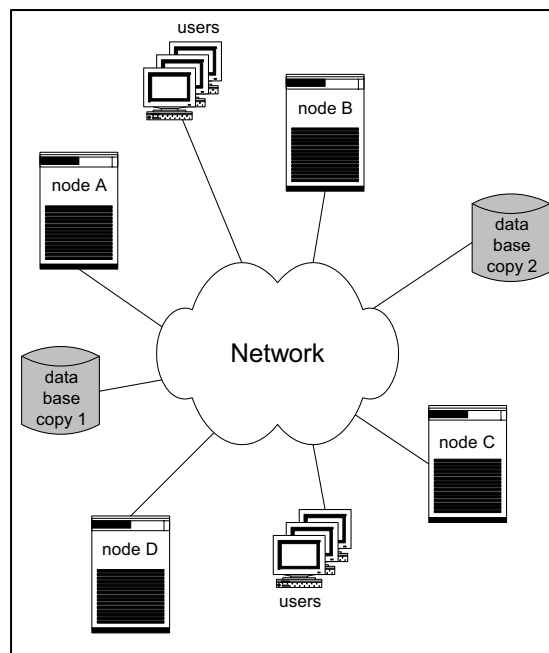
*Forward*

Let us now return to the introduction of active/active systems.

## **Achieving Extreme Availabilities**

The secret to the achievement of extreme availabilities is in the configuration. By configuring (or re-configuring) your monolithic system as an active/active architecture, the benefits described in our introduction can all be achieved.

What is an active/active system? We define it as *a network of independent processing nodes, each having access to a common replicated database. All nodes can cooperate in a common application, and users can be serviced by multiple nodes.*



**An Active/Active System**

### *Breaking the Availability Barrier – Volume 3*

Note an important implication of this definition. Active/active architectures are not just about protecting against hardware failures. In most cases, any event that will bring down a monolithic system will only bring down one node in an active/active system. Such failure events include not only hardware faults, but also software faults, operator errors, environmental failures (air conditioning, power, etc.), and manmade or natural disasters. Active/active architectures protect users against all of these faults, allowing service to be continued by simply switching users from a failed node to one or more surviving nodes.

Another implication is what active/active is not. Active/active is not a technology; it is a business solution. Active/active is not about distributed database synchronization; it is about achieving century uptimes. More specifically,

- Active/active systems are not co-located clusters. A basic tenet of active/active systems is that they protect against area-wide problems. If the nodes cannot be geographically separated, then they are not part of an active/active system.
- Active/active systems are not independent nodes using a common database. In such an architecture, the database cannot be geographically distributed and represents a single point of failure.
- Active/active systems are not those that use hardware replication for database synchronization. Hardware replication cannot guarantee referential integrity.<sup>2</sup> As a consequence, applications at synchronized sites cannot use the database copies.

---

<sup>2</sup> See Chapter 4, Volume 2, Active/Active and Related Technologies.

### *Forward*

- By the same token, active/active systems are not those that use software replication engines that do not guarantee referential integrity.
- Active/active systems are not clusters. Users on an active/active system can be put back into service in seconds by switching them to another operating node. Clusters require that another node be brought online, a process that typically takes minutes. This time delay precludes century uptimes.
- Active/active systems are not lock-stepped or voting systems because such designs require each node to process the same requests, thus precluding scalability.
- Active/active systems are not limited to enterprise applications. There are active/active distributed database systems on the market that are loosely coupled and synchronized by replication.
- Active/active systems do not require distributed disk-resident databases. Many active/active systems maintain their databases in memory.

Of course, in some cases, there may be no need for a database in an application (for example, a cluster of Web servers). In such systems, there is no context saved between operations. Implementing clusters of systems such as these is not a difficult task as it is only necessary to route any transaction to any surviving server. However, if an active database is involved such that context is retained from transaction to transaction, then providing a redundant synchronized database is necessary. This brings with it a myriad of issues. These volumes concentrate on applications which depend upon an integrated and updatable distributed database.

*Breaking the Availability Barrier – Volume 3*

In many cases, the nodes in the application network are completely symmetric. Any transaction can be routed to any node, which can read or update any set of data items in the database. Should a node fail, users at the other nodes are unaffected. Furthermore, the users at the failed node can be switched quickly to surviving nodes, with their services restored in seconds or less.

*In seconds* is the secret. Common today is the use of cluster technology to provide high availability. Should a node in the cluster fail, users are switched to a backup node. However, the applications on that node must be brought up and database tables and files opened before application services can be offered to the users. This process typically takes several minutes or more. In active/active configurations, all applications are already up and running on each node and are actively processing transactions. All that must be done is to switch over affected users to surviving nodes.

Let us say that an active/active system can recover services in three seconds and that the equivalent cluster can recover in five minutes (300 seconds). The cluster will be down one-hundred times longer than will the active/active system. This lops off two nines from the cluster's availability relative to the equivalent active/active system. A six 9s active/active system would be reduced to an availability of four 9s if it were in a cluster configuration. No wonder in 2007 many pundits still state that six 9s is not possible. But it is, as we will show in these volumes.

This leads to one of our availability rules:

**Rule 36:** *To achieve extreme reliabilities, let it fail; but fix it fast.*<sup>3</sup>

Are extreme availabilities important to you? Are the four 9s available with HP NonStop servers or with PC or Unix clusters

---

<sup>3</sup> Rules 1 through 35 are formulated in Volume 1 of this series. The complete set of rules are summarized in Appendix 1 of both Volumes 2 and 3.

## *Forward*

acceptable? As we will discuss later, surveys have shown that the costs of downtime can range from USD \$100,000 to several million dollars an hour, depending upon the application. Perhaps even worse, downtime can lead to the dreaded “CNN Moment” and massive losses in stock value (see Chapter 9, Total Cost of Ownership (TCO), in Volume 2 for what happened to AOL in 1996 and eBay in 1999). At the extreme, downtime can lead to significant property loss or even loss of life.

Only you can make this judgment. If extreme availabilities are important to your enterprise, “this book” is for you.

## **A Roadmap Through “This Book”**

As we explained earlier, “this book” is in fact Volumes 2 and 3 of our trilogy describing how to achieve extreme availabilities with active/active systems. The first volume in this series, published in 2004 by AuthorHouse ([www.authorhouse.com](http://www.authorhouse.com)) and entitled *Breaking the Availability Barrier: Survivable Systems for Enterprise Computing*, referred to herein as Volume 1, lays the groundwork and the theory supporting the concepts of active/active systems. These two current volumes focus more on the practical aspects of implementing these systems.

They are broken into four parts, Parts 1 and 2 being in Volume 2 and Parts 3 and 4 being in Volume 3:

- Part 1, Survivable Systems for Enterprise Computing, summarizes and expands on Volume 1 and provides the background for the further topics discussed in these Volumes 2 and 3. Volume 1 is not needed to understand the content or the conclusions of Volumes 2 and 3.
- Part 2, Building and Managing Active/Active Systems, demonstrates how to build the redundancy required by

### *Breaking the Availability Barrier – Volume 3*

active/active systems and how to control their cost and performance.

- Part 3, Infrastructure Case Study, describes an example of commercially available infrastructure products known to the authors to be suitable for production active/active systems. It also provides a valuable performance analysis tool for these products.
- Part 4, Active/Active Systems at Work, summarizes many of the beneficial uses of active/active systems, provides several case studies of active/active systems in use today, and describes various related technologies and issues.

The authors' intended audience for these Volumes 2 and 3 and their predecessor Volume 1 includes IT executives who feel that they must reduce the downtime of their systems, system architects and senior developers who must build these systems or modify existing systems to achieve the required availability, and operations staff who must run these systems and recover from system faults.

#### ***Part 1 – Survivable Systems for Enterprise Computing***

As the French biologist Louis Pasteur said, “Chance favors the prepared mind.” To prepare ourselves to understand active/active systems, Volume 1 of this series laid the groundwork for active/active systems and supported the concepts with mathematical analyses. As said earlier, Part 1 of this Volume 2 summarizes and expands upon the contents of Volume 1.

In Chapter 1, Achieving Century Uptimes, we talk about what is reliability and how to quantify it. We then extend these concepts to extremely reliable system configurations called *active/active* systems.

## *Forward*

Chapter 2, Reliability of Distributed Computing Systems, summarizes the mathematical foundations for active/active systems. For the reader who is mathematically adverse, you will be pleased to know that the rest of this book uses minimal mathematics (except for the data replication engine performance model, which is relegated to Appendix 2). In fact, Chapter 2 can be skipped without missing the main points of the material in the following chapters.

An overview of active/active systems is discussed in Chapter 3, An Active/Active Primer. Here we discuss in some detail the structure and characteristics of the all-important data replication engine. We also look briefly at the various failure modes and how to recover from them as well as how to control costs of active/active architectures. These later subjects are analyzed in much greater detail in Part 2 of this volume.

### ***Part 2 – Building and Managing Active/Active Systems***

The whole rationale behind active/active systems is active redundancy, which masks failures by recovering from them so rapidly that no one notices. A similar but localized philosophy is used in HP's NonStop servers, in which critical software processes are supported by backup processes in other processors resident in the same node and ready to take over in subsecond time. Also, all databases are redundant so that disk faults are masked.

There are a variety of application network topologies that have the characteristics of active/active systems. In Chapter 4, Active/Active Topologies, examples of many of these configurations are described.

In active/active systems, the inherent redundancy includes networks, databases, and processing nodes. Chapter 5, Redundant Reliable Networks, discusses ways in which to build the reliable networks needed for data replication to provide database synchronization between distributed database copies, for heartbeats to

*Breaking the Availability Barrier – Volume 3*

monitor the health of the processing nodes, and for users to be switched between nodes.

Chapter 6, Distributed Databases, describes how data replication engines can be used to keep in synchronism the multiple copies of a database in the application network. It discusses issues with replication such as data collisions and loss of data following a failure. Recovery from a failed database copy and access to a viable database copy following a node or network failure are explored.

The monitoring of a processing node's health is discussed in Chapter 7, Node Failures. A node can be considered to have failed if the processing system comprising that node has failed, if its database has failed, or if it has lost connectivity to the rest of the application network due to network faults. Techniques for recovering from a node failure are discussed, including issues such as tug-of-wars and operating in split-brain mode.

A highly beneficial use of controlled failures is shown in Chapter 8, Eliminating Planned Outages with Zero Downtime Migration (ZDM). Planned downtime is one of the major causes of reduced application availability. In many installations, the planned downtime required to upgrade a system or to execute other maintenance functions far exceeds unplanned downtime due to faults. In active/active systems, a node can be taken out of service purposefully with little or no impact on the users. This capability can be used to advantage to upgrade hardware, operating system software, application software, database structures, and so on. This technique also allows the capacity of the application network to be easily expanded by adding new nodes online.

Controlling the cost of an active/active system is as important as it is with any other system. However, active/active systems present an additional level of complexity. There are many ways to configure an active/active system to manage the appropriate compromise between

*Forward*

cost, availability, and performance. As we look at different potential configurations, how do we know which contenders are the least costly? What are the factors that enter into the total cost of ownership equation? These topics are discussed in some detail in Chapter 9, Total Cost of Ownership (TCO).

### **Part 3 – Infrastructure Case Study**

In the first two parts of “this book”, we describe why active/active systems can provide such high availability and how to build these systems. A set of tools is described that form a basis for the implementation of active/active systems. In Part 3, we look at a set of commercially available tools that fill the needs of active/active systems, and a performance model that can be used to gauge the effectiveness of such tools. The set of tools which are described are necessarily tools with which the authors are quite familiar but are otherwise reflective of several such tools in the marketplace.<sup>4</sup>

The above chapters have covered two of the three legs of the active/active triangle – availability and cost. The third leg is performance. At the heart of most active/active systems is the data replication engine, and the performance of an active/active system is directly related to this engine. In Chapter 10, Performance of Active/Active Systems, we create a performance model for a generic data replication engine and show how its various performance measures are affected by a variety of replication engine architectures. The mathematics behind the performance model are left for Appendix 2, Replication Engine Performance Model, in Volume 3.

The primary facility that is required is an appropriate data replication engine. Chapter 11, Shadowbase, describes the Shadowbase data replication engine that has been used in many such

---

<sup>4</sup> See Appendix 4, Implementing a Data Replication Project, in Volume 1 of this series, *Breaking the Availability Barrier: Survivable Systems for Enterprise Computing*, AuthorHouse, 2004.

### *Breaking the Availability Barrier – Volume 3*

implementations. Shadowbase is an example of a data replication engine with a very low replication latency (the time it takes for a change that is made to a source database to be propagated to the target database). Low replication latency is important to minimize data collisions and also to minimize data loss following a failure.

In order to take a node out of service and later return it to service, it is important to have a database copy facility that can copy the contents of an active database to a node about to be put into service (or even after it has been placed into service) while the source database is being updated. Chapter 12, SOLV, describes such a utility. Working with Shadowbase, SOLV can efficiently make a copy of an active database even while that database is being updated. In addition, future versions of SOLV will verify that two online databases are in synchronism and will resynchronize two active databases by repairing rows with differing content.<sup>5</sup>

In Chapter 13, ZDM with Shadowbase, we discuss the use of Shadowbase and SOLV to upgrade nodes in an active/active system without taking down the applications. With Zero Downtime Migrations, planned downtime can be completely eliminated since nodes in an application network can be upgraded without denying service to any user. Upgrades can include the hardware, operating system, applications, database, and networks, among others. In addition, ZDM can be used to add nodes dynamically into an application network to expand its capacity.

### ***Part 4 – Active/Active Systems at Work***

After learning how to build an active/active system and having seen an example of a tool set needed to do this, Part 4 looks at some actual uses of this technology in place today. It also describes some related technologies and issues.

---

<sup>5</sup> Check with Gravic for availability of this feature.

## *Forward*

We start in Chapter 14, Benefits of Multiple Nodes in Practice, by summarizing the various active/active system benefits that we have discussed in the book. These benefits include achieving extreme availability and very fast response time in the face of unplanned outages and even disasters, the elimination of scheduled downtime, the efficient use of all available processing capacity, the simplification of recovery testing, and application capacity expansion, both symmetric and asymmetric.

In Chapter 15, Case Studies, we look at a variety of actual uses of active/active technology. Our examples come from a wide variety of industries, including financial institutions, telecommunications, travel, web services, brokerages, plant management, and even casinos.

Finally, in Chapter 16, Related Technologies and Drivers, we explore some technologies that are related to availability. They include Grid Computing, the NonStop Server Advanced Architecture, Split Mirrors, the Real-Time Enterprise, Bulletproof Storage, and Virtual Tape. We also discuss the large number of regulatory requirements that may affect your availability decisions.

### ***Appendices***

Throughout all three volumes of this trilogy, a variety of rules applicable to highly available systems have been stated. These rules are summarized in Appendix 1, Rules of Availability. These are annotated with volume and chapter so that their context can easily be found and studied.

Appendix 1 is contained in both Volumes 2 and 3. The remaining appendices will be found in Volume 3.

Appendix 2, Replication Engine Performance Model, sets forth the detailed mathematics behind the data replication engine

### *Breaking the Availability Barrier – Volume 3*

performance model summarized in Chapter 10, Performance of Active/Active Systems. It also structures the resulting model into a set of tables suitable for creating an Excel spreadsheet for convenient performance calculations.

Appendix 3, Regulatory Requirements, summarizes the various regulatory issues that may have a bearing on the availability and operations of processing systems. These regulations are referenced in Chapter 16, Related Topics and Drivers.

Additionally, we asked a noted consultant in the field of highly available systems, Dr. Werner Alexi, President of CS Software, Concepts, and Solutions, GmbH, to provide his comments and critique on active/active systems. His views are presented in Appendix 4, A Consultant's Critique.

## **Authors' Notes**

You may have noted that this is a long book when both volumes are considered. As Winston Churchill said, “the length of this document defends it well against the risk of its being read.” To mitigate this, we would like to point out that most detail is summarized in snippets that can easily be scanned, often as rules. For instance, you might want to just hunt for the rules and read the supporting text. This will give you a good feeling for where we are trying to take you.

In many places throughout this book, reference is made to HP NonStop systems. NonStop systems were originally developed by Tandem Computers to provide very high availability. Tandem Computers was subsequently acquired by Compaq Computers, and Compaq was then acquired by HP. HP has changed the name of the Tandem systems to HP NonStop servers. The authors have considerable experience with these systems. However, concepts and recommendations presented in this book are extendable to all types of

## *Forward*

commodity systems to make them redundant, including HP Superdome, Windows Server clusters, Unix clusters, Linux servers, and IBM Parallel Sysplex systems.

Each of the chapters in this book has been written to be self-standing at the risk of some repetition. Therefore, the reader is encouraged to pick and choose the topics of interest and to read only those chapters that apply. Adequate reference is made to other chapters to suggest further reading.

## **Acknowledgements**

All three volumes of *Breaking The Availability Barrier* have benefited from reviews by many people. We gratefully acknowledge the contributions to this volume by Mary Heck for her contributions to Appendix 3 and by Dr. Werner Alexi for his critique, published in Appendix 4. We also thank Burt Liebowitz and John Carson, whose book *Multiple Processing Systems for Real-Time Applications* provided background for this work, and Jim Gray, whose many writings fueled the fire. They and others who have influenced this volume include:

Werner Alexi, CS Software  
Wendy Bartlett, HP  
Victor Berutti, Gravic  
Richard Buckle, Insession  
Robert Cline, SunGard Securities Processing  
Dan Coughlin, First Data Corp.  
Michael Crispyn, Fifth Third Bank  
Terry Cumaranatunge, Motorola  
Dick Davis, Gravic  
Giampaolo Gandini, Telecom Italia Mobile  
Jeff Glatstein, SunGard Securities Processing  
Jim Gray, Microsoft  
Jon Healy, SunGard Securities Processing

*Breaking the Availability Barrier – Volume 3*

Mary Heck, Gravic  
Tom Hoffmann, Motorola  
Bill Holenstein, Gravic  
Denise Holenstein, Gravic  
Dan Hoppmann, A. G. Edwards  
ITUG Connection staff  
Clark Jablon, Akin Gump  
Gene Jarema, Gravic  
Jim Johnson, The Standish Group  
Tim Keefauver, HP  
Rob Klotz, First Data Corp.  
Bill Knapp, Gravic  
Bob Kossler, HP  
Burt Liebowitz, Consultant  
Bob Loftis, HP  
Mike Nemerowski, SunGard Securities Processing  
Carl Niehaus, HP  
Kate Noer, SunGard Securities Processing  
Gianfranco Pompado, Telecom Italia Mobile  
Tullio Privitera, Telecom Italia Mobile  
Janice Reeder, The Sombers Group  
Steve Saltwick, HP  
Harry Scott, Carr Scott Software  
Scott Sitler, HP  
Gary Strickler, Gravic  
Bart van Leeuwen, Rabobank  
Joanne Welk, Motorola

## **About the Authors**

**Paul J. Holenstein** is Executive Vice President of Gravic, Inc., the maker of the Shadowbase line of data replication products. Shadowbase is a low latency, high-performance, real-time data replication engine that provides business continuity as well as heterogeneous data integration and synchronization. Mr. Holenstein

## *Forward*

has more than twenty-five years of experience providing architectural designs, implementations, and turnkey application development solutions on a variety of Unix, Windows, and VMS platforms, with his HP NonStop experience dating back to the NonStop I days. He was previously President of Compucon Services Corporation, a turnkey software consultancy that was acquired by Gravic. Mr. Holenstein's areas of expertise include high-availability designs, data replication technologies, disaster recovery planning, heterogeneous application and data integration, communications, and performance analysis. He has published extensively on availability topics and is a coauthor of Volume 1 of this series. Mr. Holenstein, an HP-certified Accredited Systems Engineer (ASE), earned his undergraduate degree in computer engineering from Bucknell University and a master's degree in computer science from Villanova University. He has co-founded two successful companies and holds patents in the fields of data replication, data integration, and active/active systems. He can be reached at [shadowbase@gravic.com](mailto:shadowbase@gravic.com).

**Dr. Wilbur H. (Bill) Highleyman** brings more than forty years experience in the design and implementation of real-time, mission-critical computing systems for companies such as Amtrak, Time, McGraw-Hill, Chemical Bank, Chicago Transit Authority, and Dow Jones Telerate. He is Chairman of The Sombers Group, a turnkey custom software house specializing in the development of real-time, online data processing systems with particular emphasis on fault-tolerant systems and large communications-oriented systems. In addition, he is the managing editor of the *Availability Digest* ([www.availabilitydigest.com](http://www.availabilitydigest.com)), a monthly periodical discussing topics in high availability with a focus on active/active systems. He was also Chairman of NetWeave Corporation, which developed the middleware product NetWeave. NetWeave is used to integrate heterogeneous computing systems at both the messaging and the database levels. Dr. Highleyman, a graduate of Rensselaer Polytechnic Institute and MIT, earned his doctorate in electrical engineering from Polytechnic Institute of Brooklyn. He has published

*Breaking the Availability Barrier – Volume 3*

extensively on availability, performance, middleware, and testing. He is the author of *Performance Analysis of Transaction Processing Systems*, published by Prentice-Hall, and is a coauthor of Volume 1 of this series. He holds several patents, including those in the areas of data replication and active/active systems. He can be reached at [billh@somers.com](mailto:billh@somers.com).

**Dr. Bruce D. Holenstein** is President and CEO of Gravic, Inc. Gravic's Shadowbase software supports many of the architectures described in this book and operates on systems such as Unix, Windows, NonStop, and other platforms running databases such as Oracle, Sybase, DB2, SQL Server, and SQL/MP. Dr. Holenstein began his career in software development in 1980 on a Tandem NonStop I. His fields of expertise include algorithms, mathematical modeling, availability architectures, data replication, pattern recognition systems, process control, and turnkey software. He is a coauthor of this series. Dr. Holenstein earned his undergraduate degree in Electrical Engineering from Bucknell University and his doctorate in Astrophysics from the University of Pennsylvania. Dr. Holenstein has cofounded and run three successful companies and holds patents in the field of data replication. He can be reached at [shadowbase@gravic.com](mailto:shadowbase@gravic.com).

## **Part 3 – Infrastructure Case Study**

In Part 3, we describe the architecture and characteristics of a family of available solutions for the practical implementation of active/active systems. Though there may be other products on the market today for achieving this goal, the authors are particularly familiar with the products described herein. It is intended that a thorough grounding in at least one solution family will give the reader a solid basis for evaluating various active/active system implementation products and for choosing those that are most suitable for any one specific project.

We begin this discussion with a bit of infrastructure case theory. In Chapter 10, the results of a model for predicting many of the performance attributes of an active/active solution are summarized. The details of this performance model are relegated to Appendix 2.

In Chapters 11 and 12, we describe the Shadowbase data replication engine and the SOLV online copy utility. Both are products from Gravic, Inc., based in Malvern, Pennsylvania, USA. In Chapter 13, we demonstrate how these products cooperate to implement Zero Downtime Migrations (ZDM).