



Breaking the Availability Barrier I  
Survivable Systems for Enterprise Computing

Volume 1 of 3 Volume Series  
Dr. Bruce Holenstein, Dr. Bill Highleyman, and Paul J.  
Holenstein

© 2007 Gravic, Inc. All rights reserved.

No part of this book may be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the authors.

ISBN: 978-1-4107-9231-0 (e-book)

ISBN: 978-1-4107-9232-7 (Paperback)

ISBN: 978-1-4107-9233-4 (Dust Jacket)

Library of Congress Control Number: 2003099708

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.

### **About the Authors:**

The authors of the book, Dr. Bill Highleyman, Paul J. Holenstein, and Dr. Bruce Holenstein, have a combined experience of over 90 years in the implementation of fault-tolerant, highly available computing systems. This experience ranges from the early days of custom redundant systems to today's fault-tolerant offerings from HP (NonStop) and Stratus.

Click for Book Order Information: [Breaking the Availability Barrier](#)  
or visit Amazon.com or Authorhouse.com to purchase.

## Chapter 8 - The Rules of Availability

In Part 1 of this book, we have explored basic availability concepts and have applied them to various topics, including:

- software configuration
- system replication
- system splitting
- data replication
- repair and recovery time
- failover faults
- recovery point and recovery time objectives
- ultra-high availability architectures

As we worked through this material, we generated a series of availability rules. They encapsulate in a short-cut form most of what we discussed and are summarized below. Perusing these rules is an excellent review of all that we have discussed so far.

**Rule 1:** *If all subsystems must be operational, then the availability of the system is the product of the availabilities of the subsystems.*

**Rule 2:** *Providing a backup doubles the 9s.*

**Rule 3:** *System reliability is inversely proportional to the number of failure modes.*

**Rule 4:** *Organize processors into pairs, and allocate each process pair only to a processor pair.*

**Rule 5:** *If a system can withstand the failure of  $s$  subsystems, then the probability of failure of the system is the product of the probability of failures of  $(s+1)$  systems.*

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

**Rule 6:** *System availability increases dramatically with increased sparing. Each additional level of sparing adds a subsystem's worth of 9s to the overall system availability.*

**Rule 7:** *For a single spare system, the system MTR is one-half the subsystem mtr.*

**Rule 8:** *For the case of a single spare, cutting subsystem mtr by a factor of  $k$  will reduce system MTR by a factor of  $k$  and increase the system MTBF by a factor of  $k$ , thus increasing system reliability by a factor of  $k^2$ .*

**Rule 9:** *If a system is split into  $k$  parts, the resulting system network will be more than  $k$  times as reliable as the original system and still will deliver  $(k-1)/k$  of the system capacity in the event of an outage.*

**Rule 10:** *If a system is split into  $k$  parts, the chance of losing more than  $1/k$  of its capacity is many, many times less than the chance that the single system will lose all of its capacity.*

**Rule 11:** *Minimize data replication latency to minimize data loss following a node failure.*

**Rule 12:** *Database changes generally must be applied to the target database in natural flow order to prevent database corruption.*

**Rule 13:** *Follow natural flow order when replicating so as not to create artificial activity peaks at the target database.*

**Rule 14:** *Block the ping-ponging of data changes in a bi-directional replication environment to prevent database corruption.*

**Rule 15:** *Minimize replication latency to minimize data collisions.*

**Rule 16:** *(Gray's Law) - Waits under synchronous replication become data collisions under asynchronous replication.*

**Rule 17:** *For synchronous replication, coordinated commits using data replication become more efficient relative to dual writes under a transaction manager as transactions become larger or as communication channel propagation time increases.*

**Rule 18:** *Redundant hardware systems have an availability of five to six 9s. Software and people reduce this to four 9s or less.*

**Rule 19:** (Bartlett's Law) - *When things go wrong, people get stupider.*

**Rule 20:** *Conduct periodic simulated failures to keep the operations staff trained and to ensure that recovery procedures are current.*

**Rule 21:** *System outages are predominantly caused by human and software errors.*

**Rule 22:** (Corollary to Rule 20) - *A system outage usually does not require a repair of any kind. Rather, it entails a recovery of the system.*

**Rule 23:** (Niehaus' Law) - *Change causes outages.*

**Rule 24:** *Following the failure of one subsystem, failover faults cause the system to behave as if it comprises n-1 remaining subsystems with decreased availability.*

**Rule 25:** *The possibility of failover faults erodes the availability advantages of system splitting (see Rule 9).*

**Rule 26:** (The Golden Rule) - *Design your systems for fast recovery to maximize availability, to reduce the effect of failover faults, and to take full advantage of system splitting.*

**Rule 27:** *Rapid recovery of a system outage is not simply a matter of command line entries. It is an entire business process.*

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

**Rule 28:** *RPO and RTO are both a function of the data replication technology used to maintain databases in synchronism.*

**Rule 29:** *You can have high availability, fast performance, or low cost. Pick any two.*

**Rule 30:** *A system that is down has zero performance and its cost may be incalculable.*

Following is a preview of the rules formulated in the Advanced Topics chapters, comprising Part 2 of this book.

**Rule 31:** *Minimize lock latency to minimize synchronous replication deadlocks.*

**Rule 32:** *Lock latency deadlocks under synchronous replication become collisions under asynchronous replication.*

**Rule 33:** *Designating a master node for lock coordination can eliminate lock latency deadlocks when using synchronous replication.*

**Rule 34:** *Database changes must be applied to the target database in natural flow order to maintain referential integrity. (See Rule 12).*

**Rule 35:** *A serializing facility that will restore natural flow is required following all data replication threads and before the target database in order to guarantee that the database will remain consistent and uncorrupted.*

## **Part 2 - Advanced Topics**

