

The book was found

Software Fault Tolerance Techniques And Implementation (Artech House Computing Library)



Synopsis

Look to this innovative resource for the most comprehensive coverage of software fault tolerance techniques available in a single volume. It offers you a thorough understanding of the operation of critical software fault tolerance techniques and guides you through their design, operation and performance. You get an in-depth discussion on the advantages and disadvantages of specific techniques, so you can decide which ones are best suited for your work. The book examines key programming techniques such as assertions, checkpointing, and atomic actions, and provides design tips and models to assist in the development of critical fault tolerant software that helps ensure dependable performance. From software reliability, recovery, and redundancy. . . to design and data diverse software fault tolerance techniques, this practical reference provides detailed insight into techniques that can improve the overall dependability of your software.

Book Information

Series: Artech House Computing Library

Paperback: 360 pages

Publisher: Artech House (September 30, 2001)

Language: English

ISBN-10: 1580531377

ISBN-13: 978-1580531375

Product Dimensions: 6 x 0.8 x 9 inches

Shipping Weight: 1.3 pounds (View shipping rates and policies)

Average Customer Review: 5.0 out of 5 starsÂ Â See all reviewsÂ (1 customer review)

Best Sellers Rank: #3,401,155 in Books (See Top 100 in Books) #75 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Quality Control #776 inÂ Books > Computers & Technology > Certification > CompTIA #8298 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Development

Customer Reviews

Read this book.I mean, read this book if you need to get your hands on software fault-tolerance, one of the big things that make the software you write better.The book shows you the techniques you could use so that your programs keep working in presence of failures. In few words it's all about having redundancy in software - different variants of software components doing same things which then somehow reach consensus on what's right and what's not.The basics are simple, it took me two days in the middle of reading to write a solid class framework in Python covering most (if not all)

the discussed techniques. And you know what ? It worked ! On the other hand, the basics may be simple, but the details are not. There are quite a few interesting details that you wouldn't guess beforehand. The author finds the right balance of explaining the stuff so that it's simple to understand but detailed, mathematically rigorous and technical enough. The author has her doctorate thesis on the subject. The funny thing is though, the chapter that has to do with her doctorate thesis area of research (5.3.2. Two Pass Adjudicators and Multiple Correct Results) explicitly leaves the topic unexplored. It starts like "there is 9 kinds of this stuff, let's discuss the first" and you sit deeper in your chair ready for a big thing to come, but after explaining the first flavour it suddenly goes "ok, you've had it, the rest can be found in [reference]", where reference points to the thesis. This is not bad, because you've already got the basic idea, simply amusing. The book is overly redundant to be read from it's first page to the last, and this fact is clearly mentioned in the introduction. Upon reading about a half I found myself skipping pages saying "yeah, yeah, I got it already". For example, among the discussed say 30 techniques, each can be such and such, and then such and such, leaving 4 buckets in which each individual thing can fall. Moreover, these taxonomies correlate, making just about 3 buckets. But each technique's advantages, performance problems etc. are described in full again and again, so it goes like "A is B and as such it inherits all B's pro's and con's". Simple, but it's two pages worth of text, and a large table. It may be good for reference, but not for thorough reading. Same thing with samples - the sample for a simple technique could be couple a pages of text plus a flowchart diagram plus a table with a state diagram. Very very very very very clear already. But then again, it's pleasant to have such a clear book for a change. Anyhow, if you are a software developer interested in building high quality software (as you should), and you need an introduction to software fault tolerance, this book should not be missed.

[Download to continue reading...](#)

Software Fault Tolerance Techniques and Implementation (Artech House Computing Library) Agile Systems with Reusable Patterns of Business Knowledge: A Component-Based Approach (Artech House Computing Library) Dependable Computing for Critical Applications 5 (Dependable Computing and Fault-Tolerant Systems) Software Verification and Validation: A Practitioner's Guide (Artech House Computer Library (Hardcover)) Practical Guide to Software Quality Management (Artech House Computer Science Library) Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design Broadband Networking ATM, Adh and SONET (Artech House Telecommunications Library) Convolutional Coding: Fundamentals and Applications (Artech House Communications Library) Codes for Error Control and Synchronization (Artech House Communication & Electronic Defense Library) Integrated Microwave Front-Ends with Avionics

Applications (Artech House Microwave Library (Hardcover)) Software Fault Injection MCTS Self-Paced Training Kit (Exam 70-432): Microsoft® SQL Server® 2008 - Implementation and Maintenance: Microsoft SQL Server 2008--Implementation and Maintenance (Microsoft Press Training Kit) Parallel Scientific Computing in C++ and MPI: A Seamless Approach to Parallel Algorithms and their Implementation Asap Implementation at the Speed of Business: Implementation at the Speed of Business Strategic Computing: DARPA and the Quest for Machine Intelligence, 1983-1993 (History of Computing) Wireless Computing in Medicine: From Nano to Cloud with Ethical and Legal Implications (Nature-Inspired Computing Series) Introduction to Evolutionary Computing (Natural Computing Series) CUDA Programming: A Developer's Guide to Parallel Computing with GPUs (Applications of Gpu Computing) Tiny Houses: Tiny House Plans, Woodworking on a Tiny House and Living Mortgage Free (Tiny Houses, Tiny House Living, Tiny House Plans, Small Homes, Woodworking Book 1) Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection: Obfuscation, Watermarking, and Tamperproofing for Software Protection

[Dmca](#)