

Distributed Programming With Java

As recognized, adventure as without difficulty as experience not quite lesson, amusement, as without difficulty as settlement can be gotten by just checking out a book **Distributed Programming With Java** in addition to it is not directly done, you could endure even more on this life, something like the world.

We come up with the money for you this proper as with ease as simple pretension to get those all. We give Distributed Programming With Java and numerous books collections from fictions to scientific research in any way. in the midst of them is this Distributed Programming With Java that can be your partner.

Concurrent and Distributed Computing in Java -

Vijay K. Garg 2005-01-28

Concurrent and Distributed Computing in Java addresses fundamental concepts in concurrent computing with Java examples. The book consists of two parts. The first part deals with techniques for programming in shared-memory based systems. The book covers concepts in Java such as threads, synchronized methods, waits, and notify to expose students to basic concepts for multi-threaded programming. It also includes algorithms for mutual exclusion, consensus, atomic objects, and wait-free data structures. The second part of the book deals with programming in a message-passing system. This part covers resource allocation problems, logical clocks, global property detection, leader election, message ordering, agreement algorithms, checkpointing, and message logging. Primarily a textbook for upper-level

undergraduates and graduate students, this thorough treatment will also be of interest to professional programmers.

Java in Distributed Systems - Marko Boger

2001-05-25

Large and complex software systems, such as Internet applications, depend on distributed applications. Although Java has helped reduce the complexity of distributed systems, developers still have to contend with diverse hardware platforms, remote communication over networks, and system failures. Java in Distributed Systems provides a comprehensive guide for anyone wishing to deepen their knowledge of Java in distributed applications. Beginning with a tutorial guide to distributed programming in the Java environment, it shows you how building blocks from threads to Jini can help you to fulfil Sun's vision, that 'the Network is the Computer'. It then goes on to focus on aspects that are still

challenging researchers such as concurrency, distribution, and persistence. Key Features: - One of the few books to focus specifically on Java for building distributed applications - Coverage includes threads & sockets, RMI, CORBA, Voyager, Mobile agents, JDBC, object-oriented databases, Java spaces and Jini - Includes advanced chapters on the cutting edge of Java language development, including the author's own proposed Dejay (Distributed Java), an open-source project that offers a unified approach to concurrency, distribution and persistence

Java Programming with CORBA - Gerald Brose
2001-07-05

The leading guide for Java developers who build business applications with CORBA. Acknowledged experts present advanced techniques and real-world examples for building both simple and complex programs using Java with CORBA. The authors begin with a quick overview of CORBA, Java, object request brokers (ORBs), and EJB components, then quickly move on to show how to use them to build complete Java applications. This new volume features in-depth code examples, as well as expanded coverage of cutting-edge topics, including Portable Object Adaptor (POA), Remote Method Invocation (RMI) over IIOP, and EJB.

Distributed Network Systems - Weijia Jia
2006-06-14

Both authors have taught the course of

“Distributed Systems” for many years in the respective schools. During the teaching, we feel strongly that “Distributed systems” have evolved from traditional “LAN” based distributed systems towards “Internet based” systems. Although there exist many excellent textbooks on this topic, because of the fast development of distributed systems and network programming/protocols, we have difficulty in finding an appropriate textbook for the course of “distributed systems” with orientation to the requirement of the undergraduate level study for today’s distributed technology. Specifically, from - to-date concepts, algorithms, and models to implementations for both distributed system designs and application programming. Thus the philosophy behind this book is to integrate the concepts, algorithm designs and implementations of distributed systems based on network programming. After using several materials of other textbooks and research books, we found that many texts treat the distributed systems with separation of concepts, algorithm design and network programming and it is very difficult for students to map the concepts of distributed systems to the algorithm design, prototyping and implementations. This book intends to enable readers, especially postgraduates and senior undergraduate level, to study up-to-date concepts, algorithms and network programming skills for building modern distributed systems. It

enables students not only to master the concepts of distributed network system but also to readily use the material introduced into implementation practices.

Distributed Computing in Java 9 - Raja

Malleswara Rao Pattamsetti 2017-06-30

Explore the power of distributed computing to

write concurrent, scalable applications in Java

About This Book Make the best of Java 9

features to write succinct code Handle large

amounts of data using HPC Make use of AWS

and Google App Engine along with Java to

establish a powerful remote computation system

Who This Book Is For This book is for basic to

intermediate level Java developers who is aware

of object-oriented programming and Java basic

concepts. What You Will Learn Understand the

basic concepts of parallel and distributed

computing/programming Achieve performance

improvement using parallel processing,

multithreading, concurrency, memory sharing, and

hpc cluster computing Get an in-depth

understanding of Enterprise Messaging concepts

with Java Messaging Service and Web Services

in the context of Enterprise Integration Patterns

Work with Distributed Database technologies

Understand how to develop and deploy a

distributed application on different cloud platforms

including Amazon Web Service and Docker CaaS

Concepts Explore big data technologies

Effectively test and debug distributed systems

Gain thorough knowledge of security standards

for distributed applications including two-way

Secure Socket Layer In Detail Distributed

computing is the concept with which a bigger

computation process is accomplished by splitting

it into multiple smaller logical activities and

performed by diverse systems, resulting in

maximized performance in lower infrastructure

investment. This book will teach you how to

improve the performance of traditional

applications through the usage of parallelism and

optimized resource utilization in Java 9. After a

brief introduction to the fundamentals of

distributed and parallel computing, the book

moves on to explain different ways of

communicating with remote systems/objects in a

distributed architecture. You will learn about

asynchronous messaging with enterprise

integration and related patterns, and how to

handle large amount of data using HPC and

implement distributed computing for databases.

Moving on, it explains how to deploy distributed

applications on different cloud platforms and self-

contained application development. You will also

learn about big data technologies and understand

how they contribute to distributed computing. The

book concludes with the detailed coverage of

testing, debugging, troubleshooting, and security

aspects of distributed applications so the

programs you build are robust, efficient, and

secure. Style and approach This is a step-by-step

practical guide with real-world examples.

Secure Internet Programming - Jan Vitek

2007-10-23

Large-scale open distributed systems provide an infrastructure for assembling global applications on the basis of software and hardware

components originating from multiple sources.

Open systems rely on publicly available standards to permit heterogeneous components to interact.

The Internet is the archetype of a large-scale open distributed system; standards such as HTTP, HTML, and XML, together with the widespread adoption of the Java language, are the cornerstones of many distributed systems.

This book surveys security in large-scale open distributed systems by presenting several classic papers and a variety of carefully reviewed contributions giving the results of new research and development. Part I provides background requirements and deals with fundamental issues in trust, programming, and mobile computations in large-scale open distributed systems. Part II contains descriptions of general concepts, and Part III presents papers detailing implementations of security concepts.

Concurrent Programming in Java - Douglas Lea

2000

Software -- Programming Languages.

Introduction to Reliable Distributed Programming

- Rachid Guerraoui 2006-05-01

In modern computing a program is usually

distributed among several processes. The fundamental challenge when developing reliable distributed programs is to support the cooperation of processes required to execute a common task, even when some of these processes fail.

Guerraoui and Rodrigues present an introductory description of fundamental reliable distributed programming abstractions as well as algorithms to implement these abstractions. The authors follow an incremental approach by first introducing basic abstractions in simple distributed environments, before moving to more sophisticated abstractions and more challenging environments. Each core chapter is devoted to one specific class of abstractions, covering reliable delivery, shared memory, consensus and various forms of agreement. This textbook comes with a companion set of running examples implemented in Java. These can be used by students to get a better understanding of how reliable distributed programming abstractions can be implemented and used in practice. Combined, the chapters deliver a full course on reliable distributed programming. The book can also be used as a complete reference on the basic elements required to build reliable distributed applications.

Implementing Distributed Systems with Java and CORBA - Markus Aleksy 2005-06-22

This book provides graduate students and practitioners with knowledge of the CORBA

standard and practical experience of implementing distributed systems with CORBA's Java mapping. With tested code examples that will run immediately!

Distributed Systems Architecture - Arno Puder
2011-04-18

Middleware is the bridge that connects distributed applications across different physical locations, with different hardware platforms, network technologies, operating systems, and programming languages. This book describes middleware from two different perspectives: from the viewpoint of the systems programmer and from the viewpoint of the applications programmer. It focuses on the use of open source solutions for creating middleware and the tools for developing distributed applications. The design principles presented are universal and apply to all middleware platforms, including CORBA and Web Services. The authors have created an open-source implementation of CORBA, called MICO, which is freely available on the web. MICO is one of the most successful of all open source projects and is widely used by demanding companies and institutions, and has also been adopted by many in the Linux community. * Provides a comprehensive look at the architecture and design of middleware the bridge that connects distributed software applications * Includes a complete, commercial-quality open source middleware system written in

C++ * Describes the theory of the middleware standard CORBA as well as how to implement a design using open source techniques

Distributed Computing - Mei-Ling L. Liu 2004

Distributed Computing provides an introduction to the core concepts and principles of distributed programming techniques. It takes a "how-to" approach where students learn by doing.

Designed for students familiar with Java, the book covers programming paradigms, protocols, and application program interfaces (API's), including RMI, COBRA, IDL, WWW, and SOAP. Each chapter introduces a paradigm and/or protocol, and then presents the use of a DPI that illustrates the concept. The presentation uses narrative, code examples, and diagrams designed to explain the topics in a manner that is clear and concise. End-of-chapter exercises provide analytical as well as hands-on exercises to prompt the reader to practice the concepts and the use of API's covered throughout the text.

Using this text, students will understand and be able to execute, basic distributed programming techniques used to create network services and network applications, including Internet applications.

Modern Multithreading - Richard H. Carver
2005-11-28

Master the essentials of concurrent programming, including testing and debugging This textbook examines languages and libraries for

multithreaded programming. Readers learn how to create threads in Java and C++, and develop essential concurrent programming and problem-solving skills. Moreover, the textbook sets itself apart from other comparable works by helping readers to become proficient in key testing and debugging techniques. Among the topics covered, readers are introduced to the relevant aspects of Java, the POSIX Pthreads library, and the Windows Win32 Applications Programming Interface. The authors have developed and fine-tuned this book through the concurrent programming courses they have taught for the past twenty years. The material, which emphasizes practical tools and techniques to solve concurrent programming problems, includes original results from the authors' research. Chapters include: * Introduction to concurrent programming * The critical section problem * Semaphores and locks * Monitors * Message-passing * Message-passing in distributed programs * Testing and debugging concurrent programs As an aid to both students and instructors, class libraries have been implemented to provide working examples of all the material that is covered. These libraries and the testing techniques they support can be used to assess student-written programs. Each chapter includes exercises that build skills in program writing and help ensure that readers have mastered the chapter's key concepts. The source

code for all the listings in the text and for the synchronization libraries is also provided, as well as startup files and test cases for the exercises. This textbook is designed for upper-level undergraduates and graduate students in computer science. With its abundance of practical material and inclusion of working code, coupled with an emphasis on testing and debugging, it is also a highly useful reference for practicing programmers.

Distributed, Embedded and Real-time Java Systems - M. Teresa Higuera-Toledano

2012-02-07

Research on real-time Java technology has been prolific over the past decade, leading to a large number of corresponding hardware and software solutions, and frameworks for distributed and embedded real-time Java systems. This book is aimed primarily at researchers in real-time embedded systems, particularly those who wish to understand the current state of the art in using Java in this domain. Much of the work in real-time distributed, embedded and real-time Java has focused on the Real-time Specification for Java (RTSJ) as the underlying base technology, and consequently many of the Chapters in this book address issues with, or solve problems using, this framework. Describes innovative techniques in: scheduling, memory management, quality of service and communication systems supporting real-time Java applications; Includes coverage of

multiprocessor embedded systems and parallel programming; Discusses state-of-the-art resource management for embedded systems, including Java's real-time garbage collection and parallel collectors; Considers hardware support for the execution of Java programs including how programs can interact with functional accelerators; Includes coverage of Safety Critical Java for development of safety critical embedded systems.

Concurrent and Distributed Computing in Java -

Vijay K. Garg 2004-02-04

Concurrent and Distributed Computing in Java addresses fundamental concepts in concurrent computing with Java examples. The book consists of two parts. The first part deals with techniques for programming in shared-memory based systems. The book covers concepts in Java such as threads, synchronized methods, waits, and notify to expose students to basic concepts for multi-threaded programming. It also includes algorithms for mutual exclusion, consensus, atomic objects, and wait-free data structures. The second part of the book deals with programming in a message-passing system. This part covers resource allocation problems, logical clocks, global property detection, leader election, message ordering, agreement algorithms, checkpointing, and message logging. Primarily a textbook for upper-level undergraduates and graduate students, this

thorough treatment will also be of interest to professional programmers.

Programming Distributed Computing Systems -

Carlos A. Varela 2013-05-31

An introduction to fundamental theories of concurrent computation and associated programming languages for developing distributed and mobile computing systems. Starting from the premise that understanding the foundations of concurrent programming is key to developing distributed computing systems, this book first presents the fundamental theories of concurrent computing and then introduces the programming languages that help develop distributed computing systems at a high level of abstraction. The major theories of concurrent computation—including the π -calculus, the actor model, the join calculus, and mobile ambients—are explained with a focus on how they help design and reason about distributed and mobile computing systems. The book then presents programming languages that follow the theoretical models already described, including Pict, SALSA, and JoCaml. The parallel structure of the chapters in both part one (theory) and part two (practice) enable the reader not only to compare the different theories but also to see clearly how a programming language supports a theoretical model. The book is unique in bridging the gap between the theory and the practice of programming distributed computing systems. It

can be used as a textbook for graduate and advanced undergraduate students in computer science or as a reference for researchers in the area of programming technology for distributed computing. By presenting theory first, the book allows readers to focus on the essential components of concurrency, distribution, and mobility without getting bogged down in syntactic details of specific programming languages. Once the theory is understood, the practical part of implementing a system in an actual programming language becomes much easier.

Java RMI - William Grosso 2002

If you're a distributed Java or Enterprise JavaBeans programmer, then you've undoubtedly heard of Java's Remote Method Invocation (RMI). Java programmers use RMI to write efficient, fault-tolerant distributed applications with very little time or effort. Whether you're networking across a LAN or across the Internet, RMI provides Java programmers with a lightweight solution to a heavyweight problem. Java RMI contains a wealth of experience in designing and implementing applications that use Remote Method Invocation. Novice readers will quickly be brought up to speed on why RMI is such a powerful yet easy-to-use tool for distributed programming, while experts can gain valuable experience for constructing their own enterprise and distributed systems. The book also provides strategies for working with: Serialization,

Threading, The RMI registry, Sockets and socket factories, Activation, Dynamic class downloading, HTTP tunnelling, Distributed garbage collection, JNDI, CORBA. In short, a treasure trove of valuable RMI knowledge packed into one book!

Reactive Systems in Java - Clement Escoffier

2021-11-10

Reactive systems and event-driven architecture are becoming indispensable to application design, and companies are taking note. Reactive systems ensure that applications are responsive, resilient, and elastic no matter what failures or errors may be occurring, while event-driven architecture offers a flexible and composable option for distributed systems. This practical book helps Java developers bring these approaches together using Quarkus 2.x, the Kubernetes-native Java framework. Clement Escoffier and Ken Finnigan show you how to take advantage of event-driven and reactive principles to build robust distributed systems, reducing latency and increasing throughput, particularly in microservices and serverless applications. You'll also get a foundation in Quarkus to help you create true Kubernetes-native applications for the cloud. Understand the fundamentals of reactive systems and event-driven architecture Learn how to use Quarkus to build reactive applications Combine Quarkus with Apache Kafka or AMQP to build reactive systems Develop microservices that utilize messages with Quarkus for use in event-

driven architectures Learn how to integrate external messaging systems, such as Apache Kafka, with Quarkus Build applications with Quarkus using reactive systems and reactive programming concepts

Principles of Concurrent and Distributed Programming - M. Ben-Ari 2006

Principles of Concurrent and Distributed Programming provides an introduction to concurrent programming focusing on general principles and not on specific systems. Software today is inherently concurrent or distributed - from event-based GUI designs to operating and real-time systems to Internet applications. This edition is an introduction to concurrency and examines the growing importance of concurrency constructs embedded in programming languages and of formal methods such as model checking.

Creating Components - Charles W. Kann
2003-07-28

Concurrency is a powerful technique for developing efficient and lightning- fast software. For instance, concurrency can be used in common applications such as online order processing to speed processing and ensure transaction reliability. However, mastering concurrency is one of the greatest challenges for both new and veteran programmers. Software developers with all levels of experience can refer to *Creating Components: Object Oriented, Concurrent, and Distributed Computing in Java* to

better understand how concurrency works, more effectively deploy it in program components, and reuse these components to improve program design, quality, and performance. This text introduces concurrent and component programming to students, engineers, and programmers who are familiar with Java and procedural and GUI programming. It helps them to understand and apply concurrency in Java component programming, while exploring distributed program implementation, Java threads, objects, interfaces, exceptions, component reuse, and system design and management. By providing the fundamental concepts of object-oriented components and offering templates for distributed program components, this valuable resource reveals how programmers can apply concurrency and components to solve complex problems.

Introduction to Reliable and Secure Distributed Programming - Christian Cachin 2011-02-11

In modern computing a program is usually distributed among several processes. The fundamental challenge when developing reliable and secure distributed programs is to support the cooperation of processes required to execute a common task, even when some of these processes fail. Failures may range from crashes to adversarial attacks by malicious processes. Cachin, Guerraoui, and Rodrigues present an introductory description of fundamental distributed

programming abstractions together with algorithms to implement them in distributed systems, where processes are subject to crashes and malicious attacks. The authors follow an incremental approach by first introducing basic abstractions in simple distributed environments, before moving to more sophisticated abstractions and more challenging environments. Each core chapter is devoted to one topic, covering reliable broadcast, shared memory, consensus, and extensions of consensus. For every topic, many exercises and their solutions enhance the understanding. This book represents the second edition of "Introduction to Reliable Distributed Programming". Its scope has been extended to include security against malicious actions by non-cooperating processes. This important domain has become widely known under the name "Byzantine fault-tolerance".

Designing Distributed Systems - Brendan Burns

2018-02-20

Without established design patterns to guide them, developers have had to build distributed systems from scratch, and most of these systems are very unique indeed. Today, the increasing use of containers has paved the way for core distributed system patterns and reusable containerized components. This practical guide presents a collection of repeatable, generic patterns to help make the development of reliable distributed systems far more approachable and

efficient. Author Brendan Burns—Director of Engineering at Microsoft Azure—demonstrates how you can adapt existing software design patterns for designing and building reliable distributed applications. Systems engineers and application developers will learn how these long-established patterns provide a common language and framework for dramatically increasing the quality of your system. Understand how patterns and reusable components enable the rapid development of reliable distributed systems. Use the side-car, adapter, and ambassador patterns to split your application into a group of containers on a single machine. Explore loosely coupled multi-node distributed patterns for replication, scaling, and communication between the components. Learn distributed system patterns for large-scale batch data processing covering work-queues, event-based processing, and coordinated workflows.

Nonsequential and Distributed Programming with Go - Christian Maurer

2021-01-19

Der Band bietet eine kompakte Einführung in die Nichtsequentielle Programmierung als gemeinsamen Kern von Vorlesungen über Betriebssysteme, Verteilte Systeme, Parallele Algorithmen, Echtzeitprogrammierung und Datenbanktransaktionen. Basiskonzepte zur Synchronisation und Kommunikation nebenläufiger Prozesse werden systematisch dargestellt: Schlösser, Semaphore, Monitore,

lokaler und netzweiter Botschaftenaustausch. Die Algorithmen sind in der Programmiersprache Google Go formuliert, mit der viele Synchronisationskonzepte ausgedrückt werden können.

Fundamentals of Distributed Object Systems -

Zahir Tari 2004-04-07

Distributed Object Computing teaches readers the fundamentals of CORBA, the leading architecture for design of software used in parallel and distributed computing applications. Since CORBA is based on open standards, it is the only effective way to learn object-oriented programming for distributed systems. This language independent book allows material to be taught using Java, C++ or other Object Oriented Programming Languages.

Distributed Programming with Java - Qusay H.

Mahmoud 2000

For programmers already familiar with Java, this book offers new techniques on how to develop distributed applications. Although it discusses four paradigms--low-level Sockets, Remote Method Invocation, CORBA, and Mobile Agents--this book does not favor any one of these technologies. It also allows the reader to judge the easiest approach for a particular domain of applications.

Foundations of Multithreaded, Parallel, and

Distributed Programming - Gregory R. Andrews

2000

Foundations of Multithreaded, Parallel, and

Distributed Programming covers, and then applies, the core concepts and techniques needed for an introductory course in this subject. Its emphasis is on the practice and application of parallel systems, using real-world examples throughout. Greg Andrews teaches the fundamental concepts of multithreaded, parallel and distributed computing and relates them to the implementation and performance processes. He presents the appropriate breadth of topics and supports these discussions with an emphasis on performance. Features Emphasizes how to solve problems, with correctness the primary concern and performance an important, but secondary, concern Includes a number of case studies which cover such topics as pthreads, MPI, and OpenMP libraries, as well as programming languages like Java, Ada, high performance Fortran, Linda, Occam, and SR Provides examples using Java syntax and discusses how Java deals with monitors, sockets, and remote method invocation Covers current programming techniques such as semaphores, locks, barriers, monitors, message passing, and remote invocation Concrete examples are executed with complete programs, both shared and distributed Sample applications include scientific computing and distributed systems 0201357526B04062001

Java Network Programming and Distributed

Computing - David Reilly 2002

Java's rich, comprehensive networking interfaces

make it an ideal platform for building today's networked, Internet-centered applications, components, and Web services. Now, two Java networking experts demystify Java's complex networking API, giving developers practical insight into the key techniques of network development, and providing extensive code examples that show exactly how it's done. David and Michael Reilly begin by reviewing fundamental Internet architecture and TCP/IP protocol concepts all network programmers need to understand, as well as general Java features and techniques that are especially important in network programming, such as exception handling and input/output. Using practical examples, they show how to write clients and servers using UDP and TCP; how to build multithreaded network applications; and how to utilize HTTP and access the Web using Java. The book includes detailed coverage of server-side application development; distributed computing development with RMI and CORBA; and email-enabling applications with the powerful JavaMail API. For all beginning to intermediate Java programmers, network programmers who need to learn to work with Java.

Java P2P Unleashed - Robert Flenner 2003

"Java P2P Unleashed" provides a single source for Java developers who want to develop P2P systems. The book explains the benefits of each technology and shows how to fit the P2P "pieces" together - both in building new systems and

integrating with existing ones. starts with a discussion of the P2P architecture, referencing similarities with existing, familiar systems while previewing several types of P2P applications. It explains how to plan ahead for security, routing, performance and other issues when developing a P2P application. Each technology included in the book - JXTA, Jini, JavaSpaces, J2EE, Web services - is approached from a P2P perspective, focusing on implementation concerns Java developers will face while using them. The last section includes several large-scale examples of different P2P applications - managing content, building communities, integrating services, routing messages, and using intelligent agents to gather information. The final chapter looks ahead to future developments in Java P2P technologies.

Java Distributed Computing - Jim Farley

1998-01-01

Distributed computing and Java go together naturally. As the first language designed from the bottom up with networking in mind, Java makes it very easy for computers to cooperate. Even the simplest applet running in a browser is a distributed application, if you think about it. The client running the browser downloads and executes code that is delivered by some other system. But even this simple applet wouldn't be possible without Java's guarantees of portability and security: the applet can run on any platform, and can't sabotage its host. Of course, when we

think of distributed computing, we usually think of applications more complex than a client and server communicating with the same protocol. We usually think in terms of programs that make remote procedure calls, access remote databases, and collaborate with others to produce a single result. Java Distributed Computing discusses how to design and write such applications. It covers Java's RMI (Remote Method Invocation) facility and CORBA, but it doesn't stop there; it tells you how to design your own protocols to build message passing systems and discusses how to use Java's security facilities, how to write multithreaded servers, and more. It pays special attention to distributed data systems, collaboration, and applications that have high bandwidth requirements. In the future, distributed computing can only become more important. Java Distributed Computing provides a broad introduction to the problems you'll face and the solutions you'll find as you write distributed computing applications. Topics covered in Java Distributed Computing: Introduction to Distributed Computing Networking Basics Distributed Objects (Overview of CORBA and RMI) Threads Security Message Passing Systems Distributed Data Systems (Databases) Bandwidth Limited Applications Collaborative Systems

Scientific Engineering of Distributed Java Applications - Nicolas Guelfi 2008-01-04

FIDJI 2004 was an international forum for

researchers and practitioners interested in the advances in, and applications of, software engineering for distributed application development.

Concerning the technologies, the workshop focused on "Java-related" technologies. It was an opportunity to present and observe the latest research, results, and ideas in these areas.

All papers submitted to this workshop were reviewed by at least two members of the International Program Committee. Acceptance was based primarily on originality and contribution. We selected, for these post-workshop proceedings, 11 papers amongst 22 submitted, a tutorial and two keynotes.

FIDJI 2004 aimed at promoting a scientific approach to software engineering. The scope of the workshop included the following topics: – design of distributed applications – development methodologies for software and system engineering – UML-based development methodologies – development of reliable and secure distributed systems – component-based development methodologies – dependability support during system life cycle – fault tolerance re-ment, evolution and decomposition – atomicity and exception handling in system development – software architectures, frameworks and design patterns for developing distributed systems – integration of formal techniques in the development process – formal analysis and grounding of modelling notation and techniques (e. g. , UML, metamodelling) –

supporting the security and dependability requirements of distributed applications in the development process – distributed software inspection – refactoring methods – industrial and academic case studies – development and analysis tools The organization of such a workshop represents an important amount of work.

Java Distributed Computing - Jim Farley

1998-01-01

Distributed computing and Java go together naturally. As the first language designed from the bottom up with networking in mind, Java makes it very easy for computers to cooperate. Even the simplest applet running in a browser is a distributed application, if you think about it. The client running the browser downloads and executes code that is delivered by some other system. But even this simple applet wouldn't be possible without Java's guarantees of portability and security: the applet can run on any platform, and can't sabotage its host. Of course, when we think of distributed computing, we usually think of applications more complex than a client and server communicating with the same protocol. We usually think in terms of programs that make remote procedure calls, access remote databases, and collaborate with others to produce a single result. *Java Distributed Computing* discusses how to design and write such applications. It covers Java's RMI (Remote

Method Invocation) facility and CORBA, but it doesn't stop there; it tells you how to design your own protocols to build message passing systems and discusses how to use Java's security facilities, how to write multithreaded servers, and more. It pays special attention to distributed data systems, collaboration, and applications that have high bandwidth requirements. In the future, distributed computing can only become more important. *Java Distributed Computing* provides a broad introduction to the problems you'll face and the solutions you'll find as you write distributed computing applications. Topics covered in *Java Distributed Computing: Introduction to Distributed Computing Networking Basics Distributed Objects (Overview of CORBA and RMI) Threads Security Message Passing Systems Distributed Data Systems (Databases) Bandwidth Limited Applications Collaborative Systems*

ECOOP 2008 - Object-Oriented Programming -

Jan Vitek 2008-07-10

It is a pleasure to present the proceedings of the 22nd European Conference on Object-Oriented Programming (ECOOP 2008) held in Paphos, Cyprus. The conference continues to serve a broad object-oriented community with a technical program spanning theory and practice and a healthy mix of industrial and academic participants. This year a strong workshop and tutorial program complemented the main technical track. We had 13 worksh

ops and tutorials, as well as the co-located Dynamic Language Symposium (DLS). Finally, the program was rounded out with a keynote by Rachid Guerraoui and a banquet speech by James Noble. As in previous years, two Dahl-Nygaard awards were selected by AITO, and for the first time, the ECOOP Program Committee gave a best paper award.

The proceedings include 27 papers selected from 138 submissions. The papers were reviewed in a single-blind process with three to five reviews per paper.

Preliminary versions of the reviews were made available to the authors a week before the PC meeting to allow for short (500 words or less) author responses.

The responses were discussed at the PC meeting and were instrumental in reaching decisions. The PC discussions followed Oscar

Nierstrasz's Champion pattern. PC papers had five reviews and were held at a higher standard.

JavaTech, an Introduction to Scientific and Technical Computing with Java - Clark S. Lindsey
2005-10-13

"JavaTech demonstrates the ease with which Java can be used to create powerful network applications and distributed computing applications. It can be used as a textbook for introductory or intermediate level programming courses, and for more advanced students and researchers who need to learn Java for a particular task. JavaTech is up to date with Java

5.0."--BOOK JACKET.

Creating Components - Charles W. Kann
2017-09-11

Concurrency is a powerful technique for developing efficient and lightning-fast software. For instance, concurrency can be used in common applications such as online order processing to speed processing and ensure transaction reliability. However, mastering concurrency is one of the greatest challenges for both new and veteran programmers. Software

Java Message Service - David A Chappell
2000-12-04

This book is a thorough introduction to Java Message Service (JMS), the standard Java application program interface (API) from Sun Microsystems that supports the formal communication known as "messaging" between computers in a network. JMS provides a common interface to standard messaging protocols and to special messaging services in support of Java programs. The messages exchange crucial data between computers, rather than between users--information such as event notification and service requests. Messaging is often used to coordinate programs in dissimilar systems or written in different programming languages. Using the JMS interface, a programmer can invoke the messaging services of IBM's MQSeries, Progress Software's SonicMQ, and other popular messaging product vendors. In addition, JMS

supports messages that contain serialized Java objects and messages that contain Extensible Markup Language (XML) pages. Messaging is a powerful new paradigm that makes it easier to uncouple different parts of an enterprise application. Messaging clients work by sending messages to a message server, which is responsible for delivering the messages to their destination. Message delivery is asynchronous, meaning that the client can continue working without waiting for the message to be delivered. The contents of the message can be anything from a simple text string to a serialized Java object or an XML document. Java Message Service shows how to build applications using the point-to-point and publish-and-subscribe models; how to use features like transactions and durable subscriptions to make an application reliable; and how to use messaging within Enterprise JavaBeans. It also introduces a new EJB type, the MessageDrivenBean, that is part of EJB 2.0, and discusses integration of messaging into J2EE.

Practical Distributed Processing - Phillip J. Brooke 2007-10-11

Distributed processing has a strong theoretical foundation, but many day-to-day practitioners make limited use of the advantages this theory can give them. The result includes unreliable systems with obscure and intermittent failures, that can cost time, money and in extreme cases,

lives. Reliable construction of distributed and concurrent systems must incorporate theory in practice. This book provides a concise presentation of the theory closely linked to the practical realization of these concepts. This highly practical presentation contains all the elements needed for a complete development of a distributed system. The book includes examples from C, Java and Eiffel, and sample code is available online.

Java: The Good Parts - Jim Waldo 2010-04-29

What if you could condense Java down to its very best features and build better applications with that simpler version? In this book, veteran Sun Labs engineer Jim Waldo reveals which parts of Java are most useful, and why those features make Java among the best programming languages available. Every language eventually builds up crud, Java included. The core language has become increasingly large and complex, and the libraries associated with it have grown even more. Learn how to take advantage of Java's best features by working with an example application throughout the book. You may not like some of the features Jim Waldo considers good, but they'll actually help you write better code. Learn how the type system and packages help you build large-scale software Use exceptions to make code more reliable and easier to maintain Manage memory automatically with garbage collection Discover how the JVM provides

portability, security, and nearly bug-free code Use Javadoc to embed documentation within the code Take advantage of reusable data structures in the collections library Use Java RMI to move code and data in a distributed network Learn how Java concurrency constructs let you exploit multicore processors

Concurrent, Real-Time and Distributed Programming in Java - Badr Benmammar
2017-12-27

This book provides an introduction to concurrent, real-time, distributed programming with Java object-oriented language support as an algorithm description tool. It describes in particular the mechanisms of synchronization (cooperative and competitive) and sharing of data (internal class, static variables) between threads in Java. He then discusses the use of Java for real-time applications. Consequently, a presentation of the RTSJ (Real Time Specification for Java) specification dedicated to the development of real-time applications in Java is also introduced in this book. Finally, a presentation of programming distributed in Java is presented in this book. We are particularly interested in communication using the TCP Sockets and high-level communication using Java Remote Method Invocation (RMI). The book also contains an annex which contains a practical set of application exercises in relation to the theme of the book. Knowledge of the Java language is a prerequisite for understanding the

book.

Distributed Programming - A. Udaya Shankar
2012-09-15

Distributed Programming: Theory and Practice presents a practical and rigorous method to develop distributed programs that correctly implement their specifications. The method also covers how to write specifications and how to use them. Numerous examples such as bounded buffers, distributed locks, message-passing services, and distributed termination detection illustrate the method. Larger examples include data transfer protocols, distributed shared memory, and TCP network sockets. Distributed Programming: Theory and Practice bridges the gap between books that focus on specific concurrent programming languages and books that focus on distributed algorithms. Programs are written in a "real-life" programming notation, along the lines of Java and Python with explicit instantiation of threads and programs. Students and programmers will see these as programs and not "merely" algorithms in pseudo-code. The programs implement interesting algorithms and solve problems that are large enough to serve as projects in programming classes and software engineering classes. Exercises and examples are included at the end of each chapter with on-line access to the solutions. Distributed Programming: Theory and Practice is designed as an advanced-level text book for students in computer science

and electrical engineering. Programmers, software engineers and researchers working in this field will also find this book useful.

Distributed, Embedded and Real-time Java

Systems - M. Teresa Higuera-Toledano

2012-02-07

Research on real-time Java technology has been prolific over the past decade, leading to a large number of corresponding hardware and software solutions, and frameworks for distributed and embedded real-time Java systems. This book is aimed primarily at researchers in real-time embedded systems, particularly those who wish to understand the current state of the art in using Java in this domain. Much of the work in real-time distributed, embedded and real-time Java has focused on the Real-time Specification for Java (RTSJ) as the underlying base technology, and consequently many of the Chapters in this book address issues with, or solve problems using, this framework. Describes innovative techniques in: scheduling, memory management, quality of service and communication systems supporting real-time Java applications; Includes coverage of multiprocessor embedded systems and parallel programming; Discusses state-of-the-art resource management for embedded systems, including Java's real-time garbage collection and parallel collectors; Considers hardware support for the execution of Java programs including how programs can interact with functional

accelerators; Includes coverage of Safety Critical Java for development of safety critical embedded systems.

Distributed Programming with Ruby - Mark Bates

2009-11-05

Complete, Hands-On Guide to Building Advanced Distributed Applications with Ruby Distributed programming techniques make applications easier to scale, develop, and deploy—especially in emerging cloud computing environments. Now, one of the Ruby community's leading experts has written the first definitive guide to distributed programming with Ruby. Mark Bates begins with a simple distributed application, and then walks through an increasingly complex series of examples, demonstrating solutions to the most common distributed programming problems. Bates presents the industry's most useful coverage of Ruby's standard distributed programming libraries, DRb and Rinda. Next, he introduces powerful third-party tools, frameworks, and libraries designed to simplify Ruby distributed programming, including his own Distribunaut. If you're an experienced Ruby programmer or architect, this hands-on tutorial and practical reference will help you meet any distributed programming challenge, no matter how complex. Coverage includes Writing robust, secure, and interactive applications using DRb—and managing its drawbacks Using Rinda to build applications with improved flexibility, fault tolerance, and

service discovery Simplifying DRb service management with RingyDingy Utilizing Starfish to facilitate communication between distributed programs and to write MapReduce functions for processin large data sets Using Politics to customize the processes running on individual server instances in a cloud computing

environment Providing reliable distributed queuing with the low-overhead Starling messaging server Implementing comprehensive enterprise messaging with RabbitMQ and Advanced Message Queuing Protocol (AMQP) Offloading heavyweight tasks with BackgroundDRb and DelayedJob