

<<分布式系统>>

图书基本信息

书名：<<分布式系统>>

13位ISBN编号：9787111397724

10位ISBN编号：711139772X

出版时间：2012-10

出版时间：机械工业出版社

作者：（英）库鲁里斯 等著

页数：389

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<分布式系统>>

内容概要

从移动电话到互联网，我们的生活越来越依赖于以无缝和透明的方式将计算机和其他设备链接在一起的分布式系统。

本书全面介绍分布式系统的设计原理和实践及其最新进展，并使用大量最新的实例研究来阐明分布式系统的设计与开发方法。

《分布式系统概念与设计(英文版.第5版)》前几版已被爱丁堡大学、伊利诺伊大学、卡内基—梅隆大学、南加州大学、得克萨斯a&m大学、多伦多大学、罗切斯特理工学院、北京大学等众多名校选用为教材。

第5版在上一版的基础上，新增了三章内容，分别介绍间接通信、分布式对象和组件、分布式系统设计(以google为例)。

<<分布式系统>>

作者简介

作者：（英国）库鲁里斯（Coulouris G.）（英国）Jean Dollimore（英国）Tim Kindberg（英国）Gordon Blair 库鲁里斯（Coulouris G.），伦敦大学皇后玛利学院荣誉教授，剑桥大学计算机实验室访问教授。

Jean Dollimore，伦敦大学皇后玛利学院的高级讲师。

Tim Kindberg，matter 2 media 创新技术公司创始人，Pervasive Media Studio 研究主任，英国布里斯托尔和美国帕洛阿尔托惠普实验室高级研究员，英国巴斯大学计算机科学系访问教授。

Gordon Blair，兰卡斯特大学计算系分布式系统教授，挪威特罗姆瑟大学兼职教授。

<<分布式系统>>

书籍目录

目录回到顶部

《分布式系统概念与设计(英文版.第5版)》

preface

1 characterization of distributed systems

1.1 introduction

1.2 examples of distributed systems

1.3 trends in distributed systems

1.4 focus on resource sharing

1.5 challenges

1.6 case study: the world wide web

1.7 summary

2 system models

2.1 introduction

2.2 physical models

2.3 architectural models

2.4 fundamental models

2.5 summary

3 networking and internetworking

3.1 introduction

3.2 types of network

3.3 network principles

3.4 internet

protocols

3.5 case studies: ethernet, wifi and bluetooth

3.6 summary

4 interprocess communication

4.1 introduction

4.2 the api for the internet protocols

4.3 external data representation and marshalling

4.4 multicast communication

4.5 network virtualization: overlay networks

4.6 case study: mpi

4.7 summary

5 remote invocation

5.1 introduction

5.2 request-reply protocols

5.3 remote procedure call

5.4 remote method invocation

5.5 case study: java rmi

5.6 summary

6 indirect communication

6.1 introduction

6.2 group communication

6.3 publish-subscribe systems

6.4 message queues

<<分布式系统>>

- 6.5 shared memory approaches
- 6.6 summary
- 7 operating system support
 - 7.1 introduction
 - 7.2 the operating system layer
 - 7.3 protection
 - 7.4 processes and threads
 - 7.5 communication and invocation
 - 7.6 operating system architecture
 - 7.7 virtualization at the operating system level
 - 7.8 summary
- 8 distributed objects and components
 - 8.1 introduction
 - 8.2 distributed objects
 - 8.3 case study: corba
 - 8.4 from objects to components
 - 8.5 case studies: enterprise javabeans and fractal
 - 8.6 summary
- 9 web services
 - 9.1 introduction
 - 9.2 web services
 - 9.3 service descriptions and idl for web services
 - 9.4 a directory service for use with web services
 - 9.5 xml security
 - 9.6 coordination of web services
 - 9.7 applications of web services
 - 9.8 summary
- 10 peer-to-peer systems
 - 10.1 introduction
 - 10.2 napster and its legacy
 - 10.3 peer-to-peer middleware
 - 10.4 routing overlays
 - 10.5 overlay case studies: pastry, tapestry
 - 10.6 application case studies: squirrel, oceanstore, ivy
 - 10.7 summary
- 11 security
 - 11.1 introduction
 - 11.2 overview of security techniques
 - 11.3 cryptographic algorithms
 - 11.4 digital signatures
 - 11.5 cryptography pragmatics
 - 11.6 case studies: needham-schroeder, kerberos, tls, 802.11
wifi
 - 11.7 summary
- 12 distributed file systems
 - 12.1 introduction
 - 12.2 file service architecture

<<分布式系统>>

- 12.3 case study: sun network file system
- 12.4 case study: the andrew file system
- 12.5 enhancements and further developments
- 12.6 summary
- 13 name services
 - 13.1 introduction
 - 13.2 name services and the domain name system
 - 13.3 directory services
 - 13.4 case study: the global name service
 - 13.5 case study: the x.500 directory service
 - 13.6 summary
- 14 time and global states
 - 14.1 introduction
 - 14.2 clocks, events and process states
 - 14.3 synchronizing physical clocks
 - 14.4 logical time and logical clocks
 - 14.5 global states
 - 14.6 distributed debugging
 - 14.7 summary
- 15 coordination and agreement
 - 15.1 introduction
 - 15.2 distributed mutual exclusion
 - 15.3 elections
 - 15.4 coordination and agreement in group communication
 - 15.5 consensus and related problems
 - 15.6 summary
- 16 transactions and concurrency control
 - 16.1 introduction
 - 16.2 transactions
 - 16.3 nested transactions
 - 16.4 locks
 - 16.5 optimistic concurrency control
 - 16.6 timestamp ordering
 - 16.7 comparison of methods for concurrency control
 - 16.8 summary
- 17 distributed transactions
 - 17.1 introduction
 - 17.2 flat and nested distributed transactions
 - 17.3 atomic commit protocols
 - 17.4 concurrency control in distributed transactions
 - 17.5 distributed deadlocks
 - 17.6 transaction recovery
 - 17.7 summary
- 18 replication
 - 18.1 introduction
 - 18.2 system model and the role of group communication
 - 18.3 fault-tolerant services

<<分布式系统>>

- 18.4 case studies of highly available services: the gossip architecture, bayou and coda
- 18.5 transactions with replicated data
- 18.6 summary
- 19 mobile and ubiquitous computing
 - 19.1 introduction
 - 19.2 association
 - 19.3 interoperation
 - 19.4 sensing and context awareness
 - 19.5 security and privacy
 - 19.6 adaptation
 - 19.7 case study: cooltown
 - 19.8 summary
- 20 distributed multimedia systems
 - 20.1 introduction
 - 20.2 characteristics of multimedia data
 - 20.3 quality of service management
 - 20.4 resource management
 - 20.5 stream adaptation
 - 20.6 case studies: tiger, bittorrent and end system multicast
 - 20.7 summary
- 21 designing distributed systems: google case study
 - 21.1 introduction
 - 21.2 introducing the case study: goggle
 - 21.3 overall architecture and design philosophy
 - 21.4 underlying communication paradigms
 - 21.5 data storage and coordination services
 - 21.6 distributed computation services
 - 21.7 summary
- references
- index

章节摘录

版权页：插图：CORBA's common data representation, which is concerned with an external representation for the structured and primitive types that can be passed as the arguments and results of remote method invocations in CORBA. It can be used by a variety of programming languages (see Chapter 8). Java's object serialization, which is concerned with the flattening and external data representation of any single object or tree of objects that may need to be transmitted in a message or stored on a disk. It is for use only by Java. XML (Extensible Markup Language), which defines a textual format for representing structured data. It was originally intended for documents containing textual self-describing structured data - for example documents accessible on the Web - but it is now also used to represent the data sent in messages exchanged by clients and servers in web services (see Chapter 9). In the first two cases, the marshalling and unmarshalling activities are intended to be carried out by a middleware layer without any involvement on the part of the application programmer. Even in the case of XML, which is textual and therefore more accessible to hand-encoding, software for marshalling and unmarshalling is available for all commonly used platforms and programming environments. Because marshalling requires the consideration of all the finest details of the representation of the primitive components of composite objects, the process is likely to be error-prone if carried out by hand. Compactness is another issue that can be addressed in the design of automatically generated marshalling procedures. In the first two approaches, the primitive data types are marshalled into a binary form. In the third approach (XML), the primitive data types are represented textually.

<<分布式系统>>

编辑推荐

<<分布式系统>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>