

<<网络分布计算与软件工程>>

图书基本信息

书名：<<网络分布计算与软件工程>>

13位ISBN编号：9787030307538

10位ISBN编号：7030307534

出版时间：2011-5

出版时间：科学出版社

作者：冯玉琳，黄涛，金蓓弘 编著

页数：283

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>

## <<网络分布计算与软件工程>>

### 内容概要

随着网络技术的发展和计算机应用的普及，软件系统的规模越来越大，复杂性越来越高。软件的体系结构和运行环境也发生了根本变化，软件工程方法和技术正面临着前所未有的新问题和新的挑战。

本书是在传统软件工程问题的基础上，集中阐述网络化软件的基本原理和技术，主要包括：软件系统建模、软件体系结构、分布计算原理、分布事务处理、分布式算法、分布式系统、网络软件基础架构平台、组件化软件工程开发及面向服务的计算等。

本书将网络分布计算与软件工程这两个主题相结合进行阐述，既有原理、技术和方法，又有典型系统介绍和分析。

本书适用于计算机学科的大学高年级本科生和研究生，可作为现代软件工程课的高级教程，而对于从事软件研究和开发的广大工程技术人员，也是一本基础性的专业参考书。

## <<网络分布计算与软件工程>>

### 作者简介

黄涛，1965年生。

1994年在中国科学技术大学获工学博士学位。

现任中国科学院软件研究所研究员。

博士生导师。

2002～2010年任科学院软件研究所副所长。

长期致力于分布计算、软件工程等领域研究，曾获中国科学院科技进步奖一等奖（2001）、高校科学技术二等奖（2001）、国家科技进步奖二等奖（2002）、中创软件人才奖（2002）、北京市科技进步奖一等奖（2009）。

金蓓弘，1999年在中国科学院软件研究所获工学博士学位。

现任中国科学院软件研究所研究员，博士生导师。

主要研究领域为分布式计算、移动和普适计算、软件工程。

冯玉琳，1942年生。

1982年在中国科学院获工学博士学位。

1983～1985年任美国斯坦福大学和卡内基梅隆大学访问学者。

1988年任中国科学技术大学教授。

1994～2002年任中国科学院软件研究所所长。

我国著名的计算机软件理论和工程技术专家，曾获中国科学院科技进步奖一等奖（2001）、中石化科技进步奖一等奖（2001）、国家科技进步奖二等奖（2002）、北京市科技进步奖一等奖（2009）以及优秀论文奖、优秀教学奖、先进个人奖等。

出版学术著作5部。

# <<网络分布计算与软件工程>>

## 书籍目录

序

前言

第1章软件工程发展概论

1.1 软件工程的目標

1.1.1 软件工程要素

1.1.2 软件工程面临的问题

1.1.3 软件生命期模型

1.2 软件开发方法

1.2.1 软件开发过程

1.2.2 结构化软件开发方法

1.2.3 面向对象软件开发方法

1.2.4 敏捷软件开发方法

1.2.5 软件复用

1.3 软件质量评价

1.3.1 软件质量标准

1.3.2 软件质量度量

1.3.3 软件质量保证

第2章软件系统建模

2.1 面向对象系统建模

2.1.1 面向对象建模方法

2.1.2 统一面向对象建模

2.2 UML: 统一建模的基础

2.2.1 UML的组成

2.2.2 标记方法

2.3 RUP: 统一建模的过程

2.3.1 RUP基本概念

2.3.2 核心工作流程

2.3.3 UML对开发过程的支持

第3章软件体系结构

3.1 软件体系结构模型

3.1.1 软件体系结构定义

3.1.2 软件体系结构模型

3.2 软件体系结构描述语言

3.2.1 体系结构描述语言设计考虑

3.2.2 体系结构描述语言实例研究

3.2.3 实用软件体系结构描述方法

3.3 软件体系结构风格

3.3.1 定义和作用

3.3.2 分层系统及其应用

3.3.3 容器系统及其应用

第4章分布计算原理

4.1 概述

4.1.1 网络分布计算

4.1.2 分布式系统

4.1.3 中间件

## <<网络分布计算与软件工程>>

### 4.2 基础模型

#### 4.2.1 进程模型

#### 4.2.2 时间模型

#### 4.2.3 状态模型

#### 4.2.4 失败模型

### 4.3 通信

#### 4.3.1 网络通信协议

#### 4.3.2 远程过程调用

#### 4.3.3 远程方法调用

#### 4.3.4 面向消息的通信

#### 4.3.5 组播通信

### 4.4 进程

#### 4.4.1 进程和线程

#### 4.4.2 进程组织

#### 4.4.3 进程迁移

### 4.5 并发控制

#### 4.5.1 概述

#### 4.5.2 互斥

#### 4.5.3 选举

#### 4.5.4 分布式死锁

### 4.6 寻址定位

#### 4.6.1 名字解析

#### 4.6.2 移动寻址

#### 4.6.3 分布式散列表

#### 4.6.4 分布式垃圾回收

### 4.7 容错

#### 4.7.1 进程复制

#### 4.7.2 数据复制

#### 4.7.3 一致性协议

## 第5章分布事务处理

### 5.1 分布事务

#### 5.1.1 概述

#### 5.1.2 事务模型

#### 5.1.3 原子提交协议

### 5.2 事务并发控制

#### 5.2.1 锁方法

#### 5.2.2 时间戳排序方法

#### 5.2.3 乐观并发控制方法

#### 5.2.4 事务恢复

### 5.3 工作流事务

#### 5.3.1 松弛事务模型

#### 5.3.2 事务工作流调度

## 第6章分布式算法

### 6.1 分布式路径路由算法

#### 6.1.1 宽度优先搜索算法

#### 6.1.2 最短路径路由算法

#### 6.1.3 互联网动态路由策略

## <<网络分布计算与软件工程>>

- 6.2 可靠性算法
  - 6.2.1 可靠通信算法
  - 6.2.2 节点故障处理算法
  - 6.2.3 拜占庭故障处理算法
- 6.3 负载分配算法
  - 6.3.1 静态负载分配算法
  - 6.3.2 动态负载分配算法
- 第7章分布式系统
  - 7.1 基于文件的分布式系统
    - 7.1.1 NFS
    - 7.1.2 xFS
    - 7.1.3 分布式文件系统比较
  - 7.2 基于对象的分布式系统
    - 7.2.1 CORBA
    - 7.2.2 Java EE
    - 7.2.3 DCOM
    - 7.2.4 NET
    - 7.2.5 分布式对象系统比较
  - 7.3 基于web的分布式系统
  - 7.4 基于消息和协同的分布式系统
    - 7.4.1 TIB
    - 7.4.2 JINI
    - 7.4.3 OnceDI
    - 7.4.4 基于消息和协同的分布式系统比较
  - 7.5 对等系统
- 第8章网络软件基础架构平台
  - 8.1 概述
  - 8.2 消息通信中间件
  - 8.3 事务处理中间件
  - 8.4 应用服务器
    - 8.4.1 微内核
    - 8.4.2 组件容器
    - 8.4.3 自适应资源重配
  - 8.5 数据集成中间件
  - 8.6 流程集成中间件
  - 8.7 服务集成中间件
    - 8.7.1 SOAP引擎
    - 8.7.2 BPEL运行支撑
  - 8.8 信息门户中间件
- 第9章组件化软件工程开发
  - 9.1 软件复用技术
    - 9.1.1 软件复用过程
    - 9.1.2 软件复用技术分类
    - 9.1.3 软件复用带来的问题
  - 9.2 基于组件的软件开发
    - 9.2.1 概述
    - 9.2.2 组件

## <<网络分布计算与软件工程>>

9.2.3 基于组件的软件开发方法

9.2.4 COTS

9.3 软件模式

9.3.1 概述

9.3.2 结构型模式

9.3.3 分布型模式

9.3.4 交互型模式

9.3.5 适应型模式

9.3.6 基于模式的复用

9.4 软件框架和产品线工程

9.4.1 软件框架

9.4.2 软件产品线工程方法

9.4.3 组件容器领域分析

9.4.4 组件容器产品线框架

第10章面向服务的计算

10.1 概念模型

10.2 Web服务技术

10.2.1 Web服务技术标准

10.2.2 Web服务通信

10.2.3 Web服务描述

10.2.4 Web服务发布和发现

10.2.5 Web服务组合

10.2.6 Web服务的元数据和语义

10.3 事务复合服务

10.3.1 松弛原子性验证

10.3.2 分布式并发控制

10.3.3 失败恢复

10.4 “软件即服务”和云计算

10.4.1 软件即服务

10.4.2 虚拟化

10.4.3 云计算

参考文献

附录A 专业词汇汉英对照表

附录B 专业词汇英汉对照表

附录C 常用英文缩略语表

## 章节摘录

版权页：插图：负载管理的策略要求按照任务请求获益值由高到低的顺序依次确定级别由高到低的负载管理措施，同一任务组所有的任务均采用相同级别的负载管理措施；获益值较高任务组的负载管理措施不会受获益值较低的任务组的影响，即重要任务的QOS保障效果不会受到非重要任务的影响。负载管理的算法如下：检查当前的负载状态是否满足调度检测条件，如果不能满足，则将目标负载状态置为空，并按照任务组获益值由高到低的顺序依次加入目标负载状态，然后检查目标负载状态是否可满足。

若可满足，继续循环检查下一任务组，否则尝试对该任务组作降级处理，并重新检查目标状态是否可满足；当降级到最低级别而目标负载仍不能得到满足时，该任务组及所有未检测的任务组均不包含在可满足调度的负载状态中，对它们采取准入控制措施。

算法结束时，目标负载的状态就是一个可满足调度的负载状态。

6.3.2 动态负载分配算法静态负载分配是以一种预分的方式把任务分配给服务器集群。

然而，服务器的工作负载在运行时随着计算的过程会不断发生变化，初始时好的映射分配可能会变坏，这就要求在运行时能根据系统状态进行动态调整，将负载从重负担的服务器上转移到轻负担或者空闲的服务器。

动态负载分配又称为负载平衡，是在运行过程中进行的负载分配决策。

动态负载分配算法所遵循的策略分为：（1）启动策略，决定由谁来启动负载平衡活动。

在发送者启动的方法中，由重负载的服务器启动负载转移进程。

在接收者启动的方法中，由轻负载的服务器启动负载转移进程。

（2）选择策略，决定一个服务器节点是否需要参与负载转移。

多数转移策略采用门槛规则，即当一个服务器的工作负载超过某个上限时，该节点的工作负载可以转移到网络中的其他服务器节点上。



## <<网络分布计算与软件工程>>

### 编辑推荐

《网络分布计算与软件工程(第2版)》：分布式计算和软件工程主题的融合，反映最新研究成果，网络化软件的基础系统架构，突出中间件的作用，网络分布计算原理、分布式系统架构以及组件化软件工程方法，原理方法和系统案例结合，是一部简明的现代软件工程高级教程。

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>