网络分布计算和软件工程

冯玉琳 黄涛 金蓓弘 编著

科学出版社
www.sciencep.com

网络分布计算和软件工程

冯玉琳　黄　涛　金蓓弘　编著

科学出版社

北　京

内 容 简 介

随着 Internet 和网络应用的发展，软件工程的规模越来越大，软件系统的复杂性也越来越高。本书旨在集中讲解当今软件工程面临的新问题以及为解决这些问题的新技术发展，主要内容包括软件系统建模、软件体系结构、网络分布计算、分布事务处理、分布式算法、分布式系统及组件化软件工程开发等。

本书将网络分布计算与软件工程相结合，从原理、技术到应用深入浅出地进行阐述，是软件工程的高级教程，不仅可作为高等学校软件专业本科生和研究生的教材，而且可作为从事软件研究和开发的广大工程技术人员的专业参考书。

图书在版编目(CIP)数据

网络分布计算和软件工程/冯玉琳，黄涛，金蓓弘编著. —北京：科学出版社，2003

ISBN 7-03-011356-X

Ⅰ. 网… Ⅱ. ①冯… ②黄… ③金… Ⅲ. 分布式计算机系统-软件工程-研究 Ⅳ. TP338.8

中国版本图书馆 CIP 数据核字(2003)第 025049 号

责任编辑：匡 敏 巴建芬 / 责任校对：钟 洋

责任印制：刘秀平 / 封面设计：陈 敬

科学出版社 出版

北京东黄城根北街16号

邮政编码：100717

http://www.sciencep.com

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2003 年 5 月第 一 版 开本：B5 (720×1000)

2003 年 5 月第一次印刷 印张：22

印数：1—2 500 字数：427 000

定价：35.00 元

(如有印装质量问题，我社负责调换〈环伟〉)

前　言

当今计算机世界有两大技术进步，具有划时代的意义，一是微机工作站的普及应用，另一是高速网络的出现。其导致的直接结果是：一个大规模的应用系统，可以由分布在网络上不同站点机的软件协同工作去完成。网络分布计算提供了跨越网络透明访问各种信息资源并协同处理的能力，是大规模网络应用的基础。由于软件本身的特殊性和多样性，在大规模软件开发时，人们几乎总是面临困难处境。分布式系统更使软件工程面临许多新的问题和挑战。本书集中讲解当今软件工程解决这些问题的新方法、新技术，书中每个重点主题都经过精心选择，并且融合了作者多年来在该领域的研究成果。

全书共分八章。第一章是软件工程发展概论，扼要介绍软件工程的基本概念和方法。第二章和第三章讲述软件工程发展的两个重点主题，即软件系统建模和软件体系结构。第四章和第五章讲述网络分布计算的基本特征，以及分布式系统的设计原理和关键技术，内容包括通信、进程、并发控制、名字服务、事务处理、容错等六个方面。第六章讲述分布式算法，本书不是分布式算法的专著，所以只安排一章介绍与分布式系统设计有关的几种常用算法。第七章结合典型系统案例，按不同分布式系统结构讲解基于文件的分布式系统、基于对象的分布式系统、基于Web的分布式系统以及基于消息和协同的分布式系统。最后一章讲解组件化软件工程开发，阐述基于组件的软件开发方法及相关主题。书末参考文献可用于对本书内容做进一步深入研究的索引。

本书内容丰富，有足够的技术深度，是一本深刻阐述和运用网络分布计算、反映软件工程最新发展的专门著作。本书不仅适用于高等学校软件专业高年级学生和研究生作为教材使用，而且对从事软件研究和软件开发的广大工程技术人员，也是一本很好的专业参考书。

本书第一、四、五、六章由冯玉琳撰写，第二、三章由金蓓弘撰写，第七、八章由黄涛撰写。全书最后由冯玉琳统编、修改和定稿。本书的大部分材料来自原始资料、研究论文和作者自己的研究成果，其中有部分章节内容取自 A. Tanenbaum、G. Coulouris、Jie Wu 等的相关著作。在本书写作过程中，得到了中国科学院软件研究所软件工程技术研发中心许多同志的支持，他们是：李巍、丁柯、陈宁江、刘绍华、钟华、宋靖宇、张文博、赵颉、魏峻、陈小芳、刘志峰、李剑、余亮、张建昌、明路、李松领等，在此一并表示感谢！

本书的创作得到了国家自然科学基金、国家基础研究发展计划项目的资助。

软件技术发展日新月异，因时间和水平所限，书中难免会出现疏忽错误之处，恳请读者批评指正。

冯玉琳 黄 涛 金蓓弘

中国科学院软件研究所

2003 年 1 月 15 日于北京

目　录

第一章　软件工程发展概论

自从软件工程名词诞生，历经三十年的研究和开发，人们深刻认识到，软件开发必须按照工程化的原理和方法来组织和实施。软件工程作为计算机科学的一个独立学科分支，已取得很大进步。但是由于软件产品本身的特殊性和多样性，特别是大规模网络应用软件的出现所带来的新问题，使得软件工程中，在如何协调合理预算、控制开发进度和保证软件质量等方面，软件开发人员面临更加困难的境地。在进一步深入讨论网络环境下的软件工程开发及分布式系统之前，本章首先概要介绍软件工程的最基本概念和原理，包括软件工程的目标、软件开发方法和软件质量评价。

1.1　软件工程的目标

软件工程是研究大规模软件制造的方法、工具和过程的工程科学。软件工程的规模是如此庞大，任何个人的聪明才智虽然重要，但不构成软件工程成败的最主要因素。软件工程要求参与活动的每一个人都能协同工作，并按照软件工程的规范和过程来做。本节在分析软件工程面临问题的基础上，讲述软件工程最基本的概念，包括软件工程的最基本要素、软件工程的目标以及软件工程生命周期等。

1.1.1　软件工程要素

数学定理的发明和推导是数学家高度智慧劳动的作品，其中严密的数学推理和奥妙，常为人们所叹服。计算机软件也是人们高度智慧劳动的作品，其中蕴涵的逻辑和计算，更是体现了现代科学技术的进步和力量。一个庞大复杂软件系统的分析和设计，必须遵循一定的科学原理和方法，并要求参与者充分发挥其精明和智慧。

软件工程是研究大规模软件制造的方法、工具和过程的工程科学。软件工程指导计算机软件的开发、运行和维护，采用工程化的概念、原理、方法和技术来开发和维护软件，并对软件开发过程进行有效的管理。

首先，软件工程是针对大规模软件制造的。何谓软件规模？一个最基本的度量参数是源代码行(LOC)。在高等学校内用于教学程序设计或软件工程课程的实例，一般都在5000代码行之内，只能算是小程序设计。按照通常的软件规模大小的尺度，1～5万行代码的软件是小规模软件，5～10万行的软件是中规模软件，

10～100 万行的软件是大规模软件，100 万行以上是超大规模软件。随着软件应用复杂性的增加，软件规模愈来愈大，度量软件规模的尺度也随之扩大。例如，超大规模软件的定义，已从 100 万代码行增加到 1000 万代码行。

数学证明主要依靠数学家个体的智慧，而软件规模如此庞大，只靠一个人的力量显然是不可能完成的。一个大规模的复杂软件，可能需要数百人，甚至数千人在几年的时间内协同工作才能完成。假设一个人一年可以开发出一万行的源代码，按照同样的工作效率，一个 1000 万行源代码规模的软件，是否集中 1000 人的力量工作一年就可以完成呢？答案是否定的。因为软件规模增加 1000 倍，软件复杂程度的增加远远超过 1000 倍。如何保证每个人完成的工作合在一起构成一个高质量的大型软件系统，这确实是一个极端复杂而困难的问题。不仅涉及到许多技术问题，诸如分析方法、设计技术、错误检测、版本控制等，而且还必须有严格而科学的过程管理。

考虑到研制一个软件系统同研制一台机器或建造一座楼房的过程有许多相似之处，可以参照机械工程、建筑工程中的一些概念来进行软件研制，用“工程化”的思想来指导解决软件研制过程中面临的困难。

正如所有的工程科学一样，软件工程自身遵循着一套科学的原理和方法。以此为指导，人们发展了一系列的软件工具来帮助工程软件的开发。但是，软件工程又与其他各种工程科学很不相同。软件是抽象的、逻辑性的产品，不是实物性的。研制和维护软件的过程非常难于控制。到目前为止，软件工程基本上尚无统一的设计标准，无准确的数量分析，无足够的可靠性保证，也无有效的维护手段，这就决定了软件的研制和开发较之其他工程项目困难得多。

软件工程包括三个要素，即方法、工具和过程。

• 软件工程方法为软件开发提供“如何做”的原理和技术。它包括了多方面的任务，如项目计划与评估、系统需求分析、软件体系结构、算法过程设计、编码、测试和维护等。软件开发和维护是一种复杂活动，必须用软件工程方法作指导才能进行。不同的软件方法导致不同的软件过程的差异。

• 软件工具为软件工程方法提供自动或半自动的支撑。方法和工具往往是同一问题的两个不同方面，方法是工具研制的先导，工具则是方法的实在体现。将各种软件工具集成起来，连同开发用的机器和网络设施，以及存放开发过程信息的工程数据库，一起形成软件工程开发环境。

• 软件过程是将软件工程的方法和工具综合运用，科学地进行软件的开发和实施，包括软件方法使用的顺序、要求交付的文档资料、为保证质量所需要的管理，以及软件开发各阶段要求完成的里程碑。

软件工程的目标就在于研究科学的软件工程方法，并与此相适应，发展高效实用的软件工具系统，从技术上和管理上保证软件工程项目实施的成功，以求用较少

的投资，在规定的工程期限内完成高质量的软件。

1.1.2　软件工程面临的问题

从19世纪60年代开始，人们就在讲“软件危机”，而且都在为解决软件危机面临的困难问题而进行坚持不懈的努力。软件工程作为一个学科方向，愈来愈受到人们的重视。然而，现在再来空谈所谓的软件危机已经显得很乏味了。读者们都该知道，软件社会已经发生了一场革命，以软件组件复用为代表，基于组件的软件工程技术正在使软件开发方式发生巨大改变。早年软件危机中提出的严重问题，许多方面已找到了可行的解决途径。然而，事实的情况并不能过分乐观，由于软件产品本身的特殊性和多样性，使得大规模软件工程开发时，在如何协调合理预算、控制进度以及满足客户需求和期望这三者之间，软件开发者们仍然面临困难境地。

第一，对软件工作量和开发成本的估计常常很不准确，实际成本比估计成本有可能高出一个数量级。随着技术的进步和生产规模的扩大，计算机硬件的价格在不断下降。相反，随着计算机软件社会需求的增长和软件人才的短缺，使得软件成本和软件价格不断上升。软件成本在计算机系统总成本中所占的比例也越来越大。在发达国家，计算机系统中软件费用的比例已提高到80%以上。但在发展中国家，人力资源成本相对较低，软件成本尚能保持在一个较低的水平。问题在于，软件开发成本的估价难以有统一的标准，发展中国家对软件成本的认知程度不足，加之市场的恶性竞争，导致软件价格可能低于它应有的开发成本，软件企业为了紧缩开发成本所采取的权宜措施最终会损害软件质量和客户利益。

第二，难于控制开发进度。通常对于一个软件任务，要根据其复杂性、工作量及进度要求来安排人力资源。例如有30人月的工作量，是否可按3个人完成需要10个月，由30个人完成需要1个月来安排呢？答案是否定的，因为这种工作量估计和安排方式对各部分工作互不相干的情况下才是适用的。软件系统的结构复杂，各部分工作关系密切，增加更多人员，往往不是缩短时间进度的最好办法，有时甚至适得其反，使工作陷入混乱。软件开发过程中遇到的各种意想不到的情况层出不穷，令软件开发过程很难保证按预定的计划实现。这种情况对于一般的其他工业产品，是难以想像的。

第三，软件产品的质量往往靠不住。花费了大量资金和人力开发出来的软件，其结果往往与预期有相当差距。软件的质量问题，与其他工业产品的质量问题有着很大的不同。软件设计人员对客户需求的理解和认识与用户的想法可能有着很大距离，因为程序人员几乎总是习惯以技术的角度去理解用户需求。一般说来，用户对他自己所想使用的软件的功能和性能在事前是难以确切地表述清楚的，即使用户本人是技术人员也不例外。从软件项目一开始，就存在软件开发人员与客户理解的概念差异。软件设计带有太多的灵活性和随意性，加之客户需求经常会发

生变化，这使软件质量控制成为一个很难解决的课题。

软件可靠性是软件质量的一个重要指标。软件可靠性是指软件系统能否在一定的环境条件下正确运行并实现所期望的结果。软件错误造成的后果会十分严重。医疗软件中的错误可能危及病人生命，银行软件的错误会使金融混乱，航天发射软件中的错误会使火箭试验失败。通常大约有40%左右的软件开发代价要花在软件编码完成之后的测试和排错上，但即使如此，也不能保证经测试的软件就没有错误。特别对网络上的大规模分布式软件系统测试是一件很复杂的事情，使得软件系统花在测试阶段的代价非常之大。为了提高软件的可靠性，就要付出足够的代价。

投入了巨大的人力和资金而开发出来的软件产品、最终却不能达到预期的目标，这在软件行业是一个普遍现象，如何控制和管理软件的质量，是软件工程面临的主要难题。

1.1.3 软件生命期模型

目前划分软件生命周期的方法有许多种，软件规模、软件类型、软件开发时使用的方法及开发环境等，都影响软件生命周期的划分。在划分软件生命期阶段时应遵循的一条基本原则，就是使各阶段的任务彼此间尽可能地相对独立，同一阶段各项任务的性质尽可能相同。为了用工程化的思想有效地管理软件生命期活动的全过程，一般可以将软件生命期划分为如下六个阶段：

1. 软件计划

软件计划的任务是确定软件开发的总目标、确定工程的可行性、理解工作范围和所花代价。软件计划应以可行性研究报告为基础，由软件人员和用户共同确立软件的功能和限制，制定软件计划任务书。

2. 软件需求分析

软件需求分析的目标是在系统模型分析的基础上，建立软件需求规格说明书。通常，用户知道必须做什么，但是并不能完整准确地表达出他们的要求，更不知道如何用计算机解决他们的问题；软件人员知道怎样用软件实现人们的要求，但对特定用户的具体要求并不完全清楚。因此，在需求分析阶段，系统分析员必须与用户密切配合，充分交流信息，建立经过用户确认的系统模型，作为以后软件设计和实现的基础。系统模型是要求开发的目标软件系统的抽象表示，不同的系统模型抽象会导致不同风格的软件需求规格说明。

软件需求分析是软件工程开发中重要的一步，也是决定性的一步。通过软件需求分析，才能把软件功能和性能的总体概念抽象描述为具体的规格说明，成为软

件人员与用户对要求开发的目标系统共同理解的一致基础。

3. 软件设计

软件设计就是从软件需求规格说明出发，形成软件的具体设计方案的过程。这个过程要求决定软件系统的体系结构，给出系统中各软件部件或模块的相互关系、数据库的运用以及每个部件模块的功能说明，还应考虑到在软件需求规格说明中对实现环境的要求，例如是单机环境还是网络环境等。

软件设计又细分为总体设计和详细设计两步。总体设计给出系统的整体结构，例如，软件系统由哪些模块组成以及模块间的关系，通常用层次图或结构图描述。详细设计给出各个模块的具体描述，还应该包含必要的细节，程序员根据它们可以独立地写出实际的程序代码。

4. 软件编码

根据详细设计的结果和目标系统的运行环境，选择合适的程序语言为每一个要求编码的软件模块编写程序。编写的程序应该结构良好且易于理解。可能有些软件模块不需要编程，可以利用现成的可复用组件，并对照设计要求进一步检查确认和客户化。

5. 软件测试

在一个软件系统的整个开发过程中，会出现一系列“信息转移”。信息转移是发生错误的根源。如在需求分析阶段，系统分析员错误地理解了用户的需求，发生了用户到系统分析员之间的信息转移错误；系统分析员在书写需求规格说明书时不能正确表达自己的思想，发生了系统分析员到文件的信息转移错误。在软件开发过程中，软件人员和用户都要参与，人的活动和交互不可能做到完美无缺，人犯错误的机会很多，所以，软件开发过程中总是不可避免地会出现错误。软件测试是对软件需求分析、设计和编码的最后复审，是保证软件质量的关键。

软件测试包括单元测试和综合测试。单元测试是根据详细设计说明，对软件模块进行详细测试。综合测试是在单元测试的基础上将各单元模块装在一起进行整体测试。如果要求开发的软件系统是更大的一个计算机系统的组成部分，在这种情况下，综合测试还包括所开发软件与其他软件系统放在一起进行的系统有效性测试。经过一系列的测试和排错，最后得到可交付运行使用的软件。

6. 软件维护

经过测试的软件仍然可能有错，加之用户需求和系统的运行环境也有可能发生变化，因此，交付运行的软件仍然需要继续完善、修改和扩充，这就是软件维护。

通常有四类维护活动，即改正性维护，诊断和改正在使用过程中新发现的软件错误；适应性维护，修改软件以适应环境的变化；完善性维护，根据用户的要求改进或扩充软件，使之更加完善；预防性维护，修改软件，为将来的维护活动预先做准备。

软件是程序以及为软件开发、使用和维护所需要的所有文档。根据这样的定义，软件不再仅仅是程序，研制软件也不仅仅是编写程序。按照软件工程化研制的要求，软件生命周期每一阶段完成确定的任务后，都要产生一定格式的文档。表1.1列出软件生命期每一阶段的基本任务及工作结果。

表 1.1 软件生命期阶段任务划分

阶 段	基本任务	工作结果
软件计划	理解工作范围	可行性研究报告，计划任务书
软件需求分析	定义用户要求	需求规格说明书
软件设计	确立软件结构	设计说明书
软件编码	编写程序	程序
软件测试	发现和排除错误	可运行的系统
软件维护	运行和管理	改进的系统

软件生命期划分为上述六个阶段，这就为工程化地研制软件提供了一个框架。但是，必须指出，实际软件系统的研制工作，不可能是直线进行的，常常存在着反复。研制人员往往需要从后面的阶段回复到前面。例如，在设计阶段发现需求规格说明书有不完整或者不精确之处，就需要回到需求分析阶段进行“再分析”，测试阶段发现了模块内部或者系统的错误，有时要回溯到设计阶段对原来的设计进行修正。

软件生命期前五个阶段，即计划、分析、设计、编码和测试，通常称之为软件的开发期。软件生命期的最后一个阶段称之为软件的维护期。在软件的整个生命周期中，维护的周期最长，工作量也非常大。

1.2 软件开发方法

软件开发过程是为制造软件产品，在软件工具支持下由软件开发人员完成的软件生命期中的一系列相关过程。由于目标软件产品的性质以及所采用的软件开发方法的不同，软件过程也常常很不相同。本节首先介绍几种常用的软件过程模型。尽管它们存在很大差异，但其基本特征元素都是共同的，包括

- 软件规约：描述软件功能及其行为。
- 软件设计和实现：生产制造符合规约要求的软件。
- 软件验证：确保软件符合客户需求。
- 软件演化：适应变化的客户需求。

软件开发方法是软件开发过程所遵循的法则和步骤。一个好的软件开发方法应能覆盖软件开发活动全过程，并且方便于在开发活动各阶段之间的过渡和演化。本节重点介绍两种代表性的软件开发方法，即结构化方法和面向对象方法，并在讲述软件复用概念的同时，介绍基于组件的开发方法。有关组件化软件开发方法的详细内容将在第八章再行详述。

1.2.1　软件开发过程

软件开发过程可以分为顺序的或反复的。在顺序的开发过程中，一旦某项任务完成，过程路径便不再返回到这个任务或在它之前完成的任务了。在反复的开发过程中，过程路径可以回到以前完成的任务，进行适当改变或调整，并使这种变动的效果在过程路径中向前传递。第三种可能的方式是结合顺序和反复模型的增量式方式。反复的和增量式的过程模型结合起来形成一种螺旋式的模型。

软件过程模型是对一个软件过程的抽象描述。每个过程模型从一个特定的角度描述了一个软件过程，这里介绍几种使用较普遍的过程模型。

1. 顺序的过程模型(图 1.1)

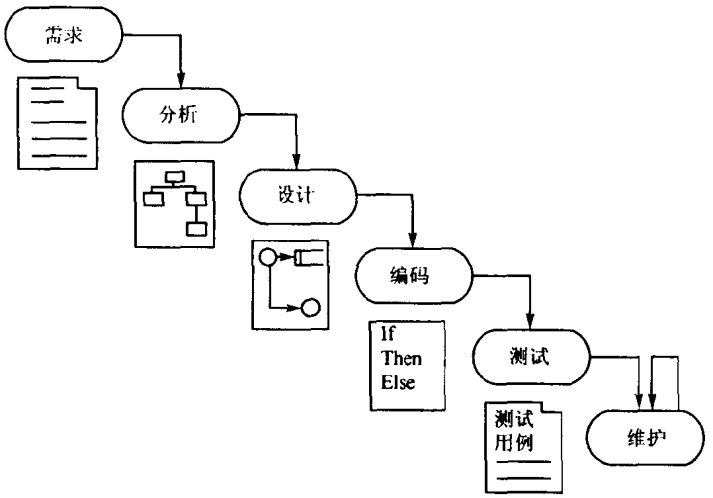

图 1.1　顺序的过程模型:瀑布型

软件工程管理人员通常最偏爱这种开发过程模型。顺序过程模型对过程的控制较强。这种模型可以进行流水型作业，可以对所耗资源和时间进行比较精确的估计。但是，前提必须是完全准确地理解用户需求，而且在开发过程中需求必须稳定。这两个条件，特别是后者，在应用软件开发过程中常常不能得到满足。

瀑布型应用开发在结构化分析设计中一直很受欢迎。它要求在下一阶段工作