

文章编号: 2095-2163(2022)12-0138-04

中图分类号: TP391.1

文献标志码: A

基于大数据技术的计算机网络应用软件开发方法设计

裴丽君

(大连东软信息学院, 辽宁 大连 116000)

摘要: 为增加计算机网络应用软件在开发过程中每秒处理请求的数量,本文引入计算机技术中的大数据技术以开展对计算机网络应用软件开发的设计研究。通过单任务与多任务应用结构化设计、计算机网络应用软件面向对象设计、软件内部构件选择与组成,设计一种全新的软件开发方式。实验结果表明,本文设计的计算机网络应用软件开发方式的每秒处理请求数量得到明显提升,能够有效提高软件的开发效率。

关键词: 大数据技术; 计算机网络; 应用软件; 开发技术

Design of computer network application software development method based on big data technology

PEI Lijun

(Dalian Neusoft University of Information, Dalian Liaoning 116000, China)

【Abstract】 In order to increase the number of requests processed per second in the development of computer network application software, this paper introduces the big data technology in computer technology to carry out the design and research of computer network application software development. Through the structural design of single task and multi task application, the object-oriented design of computer network application software and the selection and composition of software internal components, a new software development method is designed. The experimental results show that the number of requests processed per second of the computer network application software development method designed in this paper has been significantly improved, which can effectively improve the efficiency of software development.

【Key words】 big data technology; computer network; application software; development technology

0 引言

当前,现代技术的发展使得计算机已经与人们的日常生活、工作和学习有着更加密切的联系。随着这种依赖程度的不断加深,人们在使用计算机时已经不再只是局限于简单的计算机操作,以及由其提供的各类功能。针对这一问题,计算机网络应用软件的研发人员根据人们的需要,陆续开发出功能更多且更加全面的计算机网络应用软件,并逐步实现了对其现有功能的进一步拓展,从而满足人们对应用软件的功能需求^[1]。但由于软件开发是一项综合性较强的工作,传统思路和设计流程开发的软件在后期使用中会出现诸多问题,在一定程度上影响着计算机网络应用软件的质量和运行效率,同时也会对用户造成极大的困扰。针对这一问题,相关领域研究人员开展了深入的研究,并逐渐将对软件的开发作为研究重点,提出了规范性、易维护性等开发原则。但就当前总体情况而言,针对计算机硬件方面的研究与设计力度明显高于软件方面。所以,

相应的对硬件的投入力度更大,这一问题的存在对于计算机软件在市场当中的地位就会造成极大影响。综上所述可知,这些问题则使得当前计算机网络应用软件的设计难度得到进一步提升。为了增加计算机网络应用软件在开发过程中每秒处理请求的数量,提升软件开发效率,本文在引入大数据技术的基础上对计算机网络应用软件开发进行研究。

1 基于大数据技术的计算机网络应用软件开发设计

1.1 单任务与多任务应用结构化设计

为了确保开发的计算机网络应用软件可以在应用中达到既定效果,需要在开展相关研究前,对计算机程序中的应用软件进行单任务与并发任务的结构化同步设计^[2]。此过程中,可以认为应用软件中的程序是一个嵌入在逻辑端的外设程序,此程序可以根据前端事件的特点与需求,进行主循环模式的优化,并根据结构化事件的处理顺序,在主循环模块中进行程序段的设计,通过此种方式进行软件结构的

作者简介: 裴丽君(1987-),女,硕士,讲师,主要研究方向:软件开发、职业教育。

收稿日期: 2022-03-11

循环,不仅可以确保开发的应用软件具有操作便捷、结构简单等优势,同时也可以保证应用软件具有资源占用率低、运行效率高等优势^[3]。

为了达到上述预设的要求,可在开展相关设计前,对应用软件处理的优先级进行抢占,即当前端存在一个紧急处理程序或事件时,需要进行事件模块的调用,设置 α 为控制指令编程系数, β 为数据信息定义项, ϕ 为控制指令译码向量, A 表示事件模块的调用需求值,可得到事件模块调用函数为:

$$C = \frac{\beta}{\sqrt{\phi^2 + A^2}} \quad (1)$$

若事件模块调用函数取值大于 1,则可以进行事件模块调用,并在发生调用行为的同时,执行下一次循环,以此种方式,确保单任务在程序中执行的循环性^[4]。同时,在进行程序执行顺序的处理时,考虑到前端用于支撑程序的硬件设备可能存在缺陷。因此,可直接参照单片机运行模式,进行程序控制,并在控制程序时,调用一个单信号任务处理事件,将此事件表示为 DSP,则 DSP 的执行框架为: While (启动继续执行指令) {调用应用软件中的事件处理程序 1,调用应用软件中的事件处理程序 2,……,调用应用软件中的事件处理程序 n }。按照上述方式,进行单个计算机网络应用软件中单任务的执行。在完成对应用软件中单任务执行的研究与结构化设计后,应明确大部分软件中的任务均为多任务同步发送产生。因此,需要参照上述流程,进行应用软件中的多任务执行结构化设计。与上文表述相同的是,在执行多任务前,应先进行任务结构的划分,再根据执行的指令要求,进行模块任务的融合。与上述内容不同的是,执行并发任务的流程如图 1 所示。

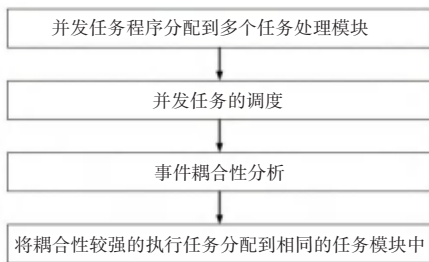


图 1 并发任务执行流程

Fig. 1 Concurrent task execution process

由图 1 可知,在执行并发任务时,不可在一个主循环程序中实施,而是要根据任务量与执行任务的指向,将并发任务程序分配到多个任务处理模块中,在实时环境中,通过 RTOS 的协调化处理,实现对并发任务的调度。在掌握多任务结构化原理后,将通

过对不同模块之间交互关系的分析与处理,进行事件耦合性的分析,将耦合性较强的执行任务分配到相同的任务模块中^[5]。在此过程中应注意的,所有待执行任务在优先级排序时,应保持任务渠道与程序之间的通信,通过对原语言的同步,减轻程序开发人员的负担,但有所区别的是,单结构化的设计对于硬件资源的要求相对较低,而多结构化的设计对于硬件资源的要求较高,因此,需要根据前端支撑硬件的赋存情况,选择是否需要增设辅助硬件的方式,进行并发任务的执行。综上所述,通过对单任务与多任务执行的并行分析,掌握不同任务在传输中的关系,将此作为依据,进行结构化层级关系的梳理,软件开发架构层级划分如图 2 所示。

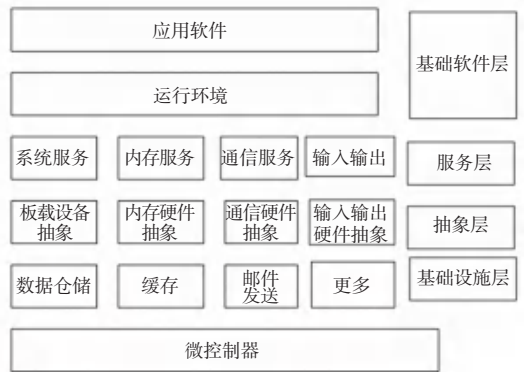


图 2 软件开发架构层级

Fig. 2 Software development architecture level

通过此种方式,便可以实现对不同类型任务应用的结构化设计。

1.2 基于大数据技术的计算机网络应用软件面向对象设计

在明确计算机网络应用软件的单任务和多任务以及相应的结构化设计后,为了确保应用软件在日后使用过程中能够更加贴合使用用户的需要,还需要引入大数据技术,对计算机网络应用软件所面向的对象进行设计。同时,在设计过程中,考虑到软件的扩展性和维护性,将数据与数据之间的封装处理操作转向在对象实体当中完成^[6]。由于处于边界的数据无法实现对内部数据的访问,并且操作十分困难,因此引入大数据技术,为对象之间的信息传递提供连接,从而实现 2 个软件用户的间接访问。同时,引入大数据技术后,计算机网络应用软件面向对象的选择能够更加贴合人们的思考方式,使得问题空间与解空间的描述更加准确,即便在不了解内部细节的情况下,也能够对软件开发的动态、重载等开发操作提供支撑条件。通过上述方式的软件开发就

能够在后期实际应用中具备更强的重用性。此外,在这一阶段,还可结合大数据技术中的对象挖掘技术实现对不同软件用户特征的挖掘。将挖掘对象每一类别下的对应特征出现概率进行统计,可引入如下公式:

$$T = \sum_{i=1}^n \lambda_i \cdot \frac{1}{n} \quad (2)$$

其中, T 表示挖掘对象每一个对应特征出现的概率; λ_i 表示挖掘对象的特征属性数据; n 表示计算机软件当中所有特征属性数据个数。

根据上述公式,确定每一类别下对应特征出现的概率,将出现概率最多的特征相对应的挖掘对象作为软件开发过程中的面向对象。并在确定面向对象后,通过软件建模、详细设计等步骤实现对面向对象的具体描述^[7]。同时,在这一过程中,也可使用 MSC 对用户实例进行建模。模型中分为 4 个模块,分别为:应用模块、面向对象运行支撑模块、PTOS 对象化封装模块和 RTOS 模块。通过各个模块之间的相互配合,以此来对计算机网络应用软件静态和动态行为做出描述。随后研究还发现,通过模型中后 2 种模块的引入还能够进一步提高计算机网络应

用软件的内存空间占用,从而有效降低其运行效率。在面向对象设计阶段,根据建模得出的结果选择应用软件的体系结构,并对高级对象进行细化,描述对象之间的通信关系,并确定被动对象,从而实现计算机网络应用软件面向对象的设计。

1.3 软件内部构件选择与组成

在完成上述设计后,下面将对计算机网络应用软件开发中的软件内部构件组成与选择进行研究。为了确保内部构件满足计算机网络应用软件的开发需求,需要在研究中引进“生产线”理念,明确规划后的构建模型与完善的软件体系结构对于软件开发的重要性,因此,要在开发中进行程序架构的设计与维护,即对软件进行“粗粒度”的维护。目前,市场上比较经典的软件架构包括顺向逻辑软件、复用软件,后者是现在较为常用的软件体系结构。在复用软件结构中,可以将每个体系构成作为一个对接口,通过此接口与外部的交互,进行服务信息的请求,相比常规的黑盒式服务软件,此种构架模式具有更强的独立性与抽象性,可以满足不同终端用户对于软件功能的个性化需求。综合上述分析,对软件构件模型与软件体系结构进行描述,结构示意图如图 3 所示。

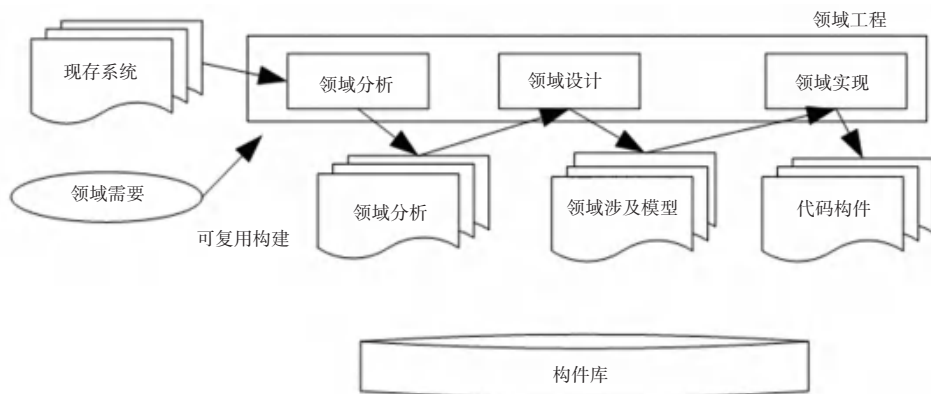


图3 构件模型与软件体系结构

Fig. 3 Component model and software architecture

按照图 3 给出的结构进行软件体系结构的设计,在设计过程中应注意的是,不同的构件与软件适配存在唯一性,即设计的软件只能在预设的空间中应用,一旦发生前端硬件的拓展,极有可能存在软件不适配的问题。因此,也可以将此构件模式称之为“构件-架构”模式。为了确保软件功能的完善性,可在进一步的设计中,引进高阶抽象理念、即 Agent 理念,基于此理念将不同层次上的构件体系进行连接,可以确保基础架构与模型架构在应用中保持同步性。在进行应用程序交付时,可通过对用户需求的集中分析,进行不同领域中构件代码的编译,并在

编译后,将所有与之相关的构件导入一个标准的构件库中,确保代码格式标准后,即可实现对计算机网络应用软件的交付。此外,可在交付前,根据软件开发需求,进行应用软件的试运行与调试,确保软件功能无误后,即可执行交付。综上所述,完成基于大数据技术的软件开发设计。

2 对比实验

本文在结合大数据技术的基础上,提出一种全新的软件开发方式,其界面如图 4 所示。



图 4 软件开发界面

Fig. 4 Software development interface

为了进一步验证设计方法在实际中的应用效果,选择将本文方法与传统的基于安全技术的开发方式进行实验对比,实验对象为:同一计算机应用功能需求软件的开发。为了方便论述,将本文方法设置为实验组,将文献[5]方法作为对照组,做了如下对比实验:

为了实现对 2 种软件开发方式的验证,选择将开发时的运行效率作为评价指标,针对实验组与对照组在实验过程中每秒处理的请求数量作为评价量化结果。为了确保实验结果的客观性,2 种开发方式下所需的开发条件均相同。选择将某一软件的开发程序编写作为实验用例,分别利用实验组和对照组这 2 种开发方式对该实验用例进行开发。在该实验用例中,包含了对项目内容的配置、对视图的渲染以及文件缓冲和程序创建等内容。在实验过程中,引入黑盒测试模式,在确保 2 种开发方式正常运行的情况下,对 5 种不同操作任务下每秒处理的请求数量进行记录。其中,操作任务分别为:直接对开发程序编写文件进行访问(任务 1);直接对视图模块进行访问(任务 2);对模块、操作和参数进行访问(任务 3);正常模式下对软件参数进行设置(任务 4);对多组参数同时设置(任务 5)。在实验组与对照组均完成上述 5 个操作任务后,将实验结果进行记录,并绘制成表 1。

表 1 实验组与对照组实验结果对比表

Tab. 1 Comparison of experimental results between experimental group and control group

操作任务	实验组每秒处理请求数量/(个·s ⁻¹)	对照组每秒处理请求数量/(个·s ⁻¹)
任务 1	1 526	625
任务 2	1 523	724
任务 3	1 059	432
任务 4	1 625	435
任务 5	1 425	238

从表 1 得出的实验数据可以看出,在完成 5 项操作任务后,实验组每秒处理请求的数量明显多于对照组,同时在实验过程中发现,实验组总共完成的请求超过 10 000 个,而对照组总共完成的请求小于 2 500 个,对照组软件开发明显不具备优势条件。因此,通过本文上述实验得出的结果能够证明,在确保最终计算机网络应用软件的开发生质量均符合标准的情况下,引入大数据技术后可以提升软件开发的运行效率,并且能够充分满足开发者的设计需要,为计算机网络应用软件在市场当中的竞争实力提升提供条件。

3 结束语

本文在引入大数据技术后,从 3 个方面对计算机网络应用软件的开发生进行了详细的设计,并在传统软件开发方式应用效果的基础上,增加了每秒处理请求数量,从而使得软件开发的效率得到了进一步提升。在今后的研究当中,还将进一步细化计算机网络应用软件的类别,针对嵌入式软件的开发和应用进行更加深入的研究,从而确保开发的软件中的各个构件和运行功能能够满足软件使用用户的需要,以期为提升软件开发效率提供一定帮助。

参考文献

- [1] 蒋岑. 大数据在区域医疗中的应用[J]. 电脑迷, 2016(12): 121.
- [2] 丁勇. 安全技术 in 计算机软件开发中的应用研究—评《计算机安全技术》[J]. 现代雷达, 2021, 43(01): 95.
- [3] 陈中凯. 试论在计算机软件开发中数据库安全设计的应用实践[J]. 信息系统工程, 2020(12): 119-120, 122.
- [4] 吴慧林. 大数据时代下软件工程方法在计算机软件开发中的实践[J]. 普洱学院学报, 2021, 37(03): 16-18.
- [5] 刘伟, 麻名蕊. 动态可重构穿戴计算机软件平台开发路径及网络通信实现分析[J]. 电子测试, 2020(03): 80-81.
- [6] 平金珍, 王茜. 计算机软件开发中数据库安全设计的应用实践分析[J]. 电子元器件与信息技术, 2020, 4(05): 34-35, 42.
- [7] 蒋岑, 吴迪. 隐蔽无线通信网络传输信息云存储密文检索[J]. 计算机仿真, 2021, 38(06): 125-128, 137.