

基于普适环境的自适应系统研究

李 华, 万晓冬

(南京航空航天大学 自动化学院, 江苏 南京 210016)

摘 要:作为新的计算模式, 普适计算为应用技术界开辟了一个崭新、宽阔的研究领域。普适计算的目标是无时无处不在而又不可见, 具有泛在性、便捷性和适应性的特点, 对软件自适应机制提出了新的挑战。针对自适应技术用于普适计算下存在的问题, 对自适应技术的体系结构展开研究, 提出了一个适用于普适计算环境的自适应中间件模型。力图实现一种能够较好支持普适计算环境的软件基础架构, 并通过分布仿真原型系统的实验比较说明该中间件体系结构的执行效率。对自适应系统与自适应策略进行了探讨。

关键词:普适计算; 自适应技术; 中间件

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2010)07-0192-04

Research on Adaptive System Based on Pervasive Environment

LI Hua, WAN Xiao-dong

(School of Automation, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: As a new model of computing, pervasive computing inaugurates a whole new and wide field of research for the application and technology kingdom. The aim of pervasive computing is "everywhere, anytime and invisible" computing. Moreover, there are three characters - ubiquitous, convenience and self - adaptive in pervasive environment, thus it brings a new challenge for the scheme of software self - adaptive. To solve the questions of adaptive technology for pervasive computing environments, research on the architecture of adaptive technology is introduced. An adaptive middleware model is presented according to the character of pervasive computing. Aimed to propose a software infrastructure that could meet requirements of pervasive computing and facilitate the development of pervasive applications. Experiments made on the prototype middleware system called 'distributed interactive simulation prototype system' showed the middleware architecture. In addition, adaptive system and adaptive strategy are discussed in detail.

Key words: pervasive computing; adaptive technology; middleware

0 引 言

普适计算最早的定义来源于 IBM 公司: 通过新一代的设备根据需求随时随地方便地访问相关信息, 并可轻松地对其进行操作。现在国内普遍采用的定义为^[1]: 普适计算是信息空间与物理空间的融合, 在这个融合的空间中人们可以随时随地透明地获得数字化的服务。普适计算是一种新型的计算模式, 在该模式下, 计算以人为中心, 人机交互类似于人与人之间的自然交流方式, 用于计算的设备无处不在, 弥漫在人们生活的环境中, 并能够便捷地为人们提供所需要的服务, 而使用计算设备的人却感知不到计算机的存在。普适计算^[2]是一种非常广泛的计算概念, 包括了无线计算、网

络计算、移动计算、日常计算, 以及嵌入式计算和实时计算等侧重点不同的内容, 其目的在于突破基于桌面系统的计算模式, 使人们能够随时随地获得计算服务。

但是在普适计算环境中, 对各种资源(含设备)的访问控制需求确是各不相同的^[3]。由于普适计算环境下各种资源设备在访问控制系统中不同的角色和地位, 整个访问控制系统对各种资源(含设备)的访问控制要求, 无论从反应速度, 策略复杂度, 还是上下文关联程度等等方面都大相径庭。另外, 资源(含设备)本身的特点与能力, 如本身的计算能力等, 对于访问控制需求也有一定的影响。

这种不均衡的访问控制需求以及高异构性的系统环境使得普适计算的访问控制^[4]不可能仅仅保持一种固定的组织结构或功能行为, 它必须能够根据不同的需求以及相应的设备特点, 进行相应的自定义化的配置, 以保持系统稳定高效的运行。

当然, 自适应问题在很大程度上依赖于普适计算

收稿日期: 2009-07-23; 修回日期: 2010-03-08

基金项目: 武器装备预研基金项目(9140A04030307HK0201)

作者简介: 李 华(1974-), 男, 硕士, 讲师, 研究方向为计算机测控; 万晓冬, 硕士, 助理研究员, 研究方向为计算机测控。

系统软件的设计与实现。而普适计算^[5]环境下的访问控制,属于位于系统软件层次之上的应用层。因此,普适计算环境下的访问控制自适应性的解决不仅仅依靠访问控制系统本身的设计与实现。

1 自适应定义

科学界对于“自适应”问题的研究由来已久,对于自适应的理解也是多方面的。最早对于“自适应”的论述出现于1971年,Ya. Z. Tsypkin^[6]在他的著作《Adaptation and Learning in Automatic Systems》中指出,自适应是自动计算的一个方面。从软件需求角度来考虑,B. Tekinerdogan^[7]认为:自适应是指系统整体或部分地适应需求的变化。P. Preizy^[8,9]等认为:自适应是软件的进化,通过改变软件自身使之在新的环境下继续正常运行。针对自适应的多种定义,浙江大学的吴卿^[10]进行了一个较为完善的总结。他认为:自适应是软件的一种能力和特性,随着运行环境上下文的变化,通过自主自动地调整软件的组织结构或是调整软件的功能行为来满足变化的要求,达到“以变应变”的目标。

目前,自适应性的研究已经成为软件工程难点和热点。专家学者普遍认为:复杂的软件系统中,重新获得灵活性和自适应性,已经成为当前软件工程中最根本的挑战;自适应系统的实现已经成为软件体系结构中一个非常重要的方面,这包括在系统设计、协作、检测、评估和实现的各个阶段实现无缝的自适应^[10]。针对普适计算这一特殊的计算形态,其复杂的软件系统设计中当然少不了对自适应性的考虑。下面,就对普适计算环境下的自适应技术相关问题进行分析与探讨。

2 自适应中间件定义

普适环境中,传统的中间件平台已不能胜任一些复杂的场合,这就迫切需要建立一种新的中间件平台。将现有的中间件平台自适应化(自适应中间件)已成为当前普适计算研究的一个热点和难点。

自适应中间件,即指中间件内部具有自适应的特性,可以随着当前外部环境的变化以及软件内部执行情况的变更来进行自适应的配置和重配置,包含组成结构上和行为功能上的动态调整,以满足环境变化的需求,从而达到中间件的“以变应变”的特性。

自适应中间件具有与传统中间件相同的本质特性,均为分布式仿真应用开发提供统一的编程模型,屏蔽底层开发应用平台的异构性等细节。不同之处在于:自适应中间件具有开放性、动态性、可重配置性、智能性和可靠性等特征。

3 自适应系统结构

自适应系统结构如下:

(1)自适应目标:指从哪一方面向系统提供自适应,像任务实时性、网络传输延迟、流媒体连续性等。例如:任务目标为实时性,可以通过CPU资源的动态调整来实现。

(2)自适应策略:为系统提供自适应所采取的方法和手段,像任务再分配策略、反馈控制策略等。这些策略一般具有通用性,适用于多个自适应目标。

(3)自适应机制:为实现自适应策略所需的基本措施,像可抢占机制、基于优先级的分配机制等。

(4)自适应OS^[11]:为了使系统能对环境变化作出自适应处理,需要在OS(操作系统)中添加自适应处理模块,满足对环境资源的自适应需要。图1所示为电子科技大学研究的嵌入式实时操作系统Delta OS系统体系结构图,具有自适应功能。

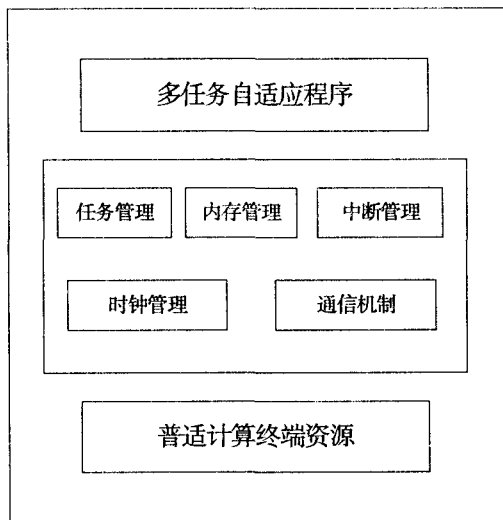


图1 Delta OS系统体系结构

(5)系统状态信息:即系统在进行CPU资源管理时所需的系统参数,像CPU利用率、实时任务的截止时间错过率等。

(6)自适应策略的实现:从算法和编程上实现某种具体的自适应策略。

4 自适应系统实现

在CORBA中,开发人员先用IDL定义一组接口,并用本地语言实现这些接口,然后由IDL编译器生成相应Stubs和Skeletons,为对象之间的调用提供所需的基础设施,Stubs和Skeletons分别运行于客户端和服务端。类似,可以应用CORBA编程规范来实现自适应系统。首先用IDL分别为自适应系统定制代理、功能代理、系统状态对象、适配器以及系统自适应模块定

义接口;然后用 IDL 编译器生成相应 Stubs 和 Skeletons,并用 C++ 语言实现各接口。此外,自适应系统的实现还包括合同、系统状态对象、自适应策略以及绑定各对象等工作。

开发人员首先根据应用程序需求,用 C++ 语言编写一个适配器对象(AdapterConfig.C),用于将所需要的自适应策略绑定到相应功能代理上,即指定系统提供的自适应策略及相应接口;在此基础上,另外设计一个封装对象(Wrapper.C),用于初始化自适应系统内部各对象,定位与设置服务对象。当适配器对象和封装对象均被实例化以后,自适应系统即可通过适配接口向应用程序提供自适应服务。如图 2 所示。

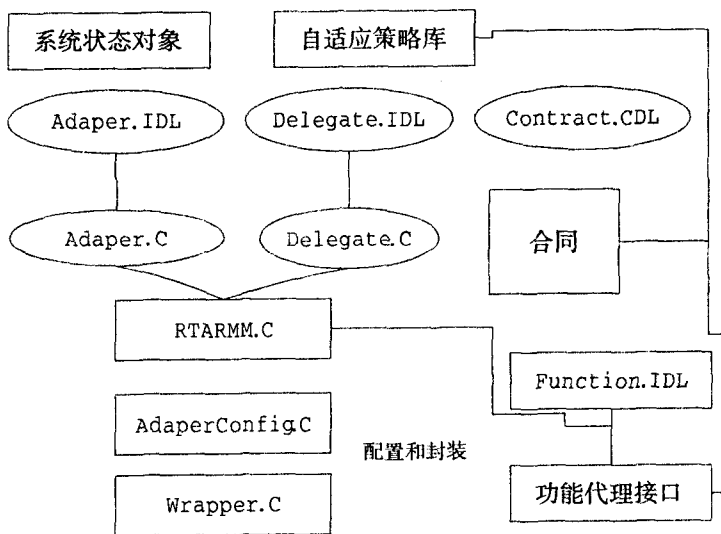


图 2 配置自适应系统

5 自适应策略

普适计算中,终端所处的网络环境是动态变化的,可获得的计算资源也是变化的,其调度策略也要求是动态的或者自适应的^[12]。目前传统的研究方向主要集中在如何确保任务的实时性和确定性上,即通过建立数学模型对实时任务的可调度性进行静态分析,并利用任务调度、资源预留、许可控制等策略来保证任务按确定方式运行。因此,以往绝大部分实时调度(CPU 资源分配)算法都是围绕着实时系统,并假设任务都有确定的执行时间、周期和截止时间,任何错过截止时间的任务都将导致系统失效。为此,学术界提出了一些新的自适应调度策略,这些策略认为实时任务在资源消耗上具有一定灵活性,可以在运行时动态地调整任务。此外,绝大多数策略都采用了两级调度结构,顶层算法在系统载荷发生变化时调整任务,底层算法则进行实际的任务调度。两级调度使得自适应策略更加灵活,软件实现更加简单。因此在中间件层实现自适应

策略可能是一种更好的选择。

自适应策略是实现动态配置的前提。它主要负责两个方面的内容:其一是通过上下文感知机制对外界环境和内部资源的分析,在保证系统安全可靠的同时,选择合适的自适应策略。其二是评价自适应结果的好坏,如果自配置后的结果不能满足应用程序的需求,并且不优于先前的配置时,自适应策略应该能够返回上次的配置。这样就保证了系统不会运行的比原来的差。

6 普适环境自适应中间件模型探讨

在普适计算环境中,把自适应机制放在中间件层,它不仅方便了应用程序的开发与应用,而且增加了应用程序对环境的自适应性。根据自适应中间件对环境感知的要求以及自适应中间件本身的特点,提出了一种能够适应环境变化的自适应中间件模型,这种模型能够使自适应策略的运用更加灵活,应用程序对环境的自适应性更强。根据功能的不同,将自适应中间件模型分为五个模块:Application、上下文感知、自适应执行体、自适应机制、Sensor。

其结构如图 3 所示。

●模块简介:

1) Application。

获取上下文信息并发送上下文信息给上下文感知应用程序,进而用于开发更为复杂的上下文感知应用。

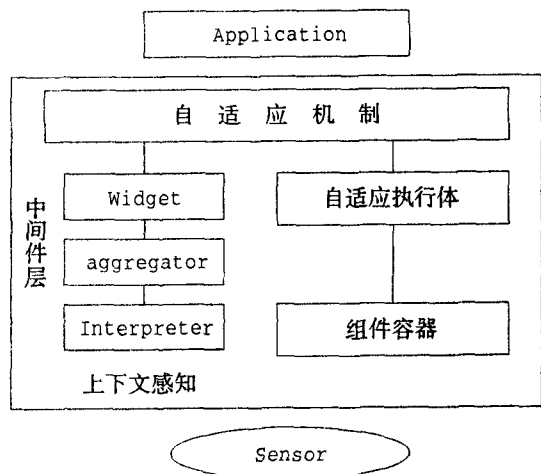


图 3 自适应中间件结构

2) 上下文感知。

上下文感知是由三种抽象组成的:widgets, aggregators 和 interpreters。其中, Context widgets 封装了单个上下文的信息, aggregators 把许多 widgets 组合起来

提供一个高层 widgets, interpreters 可以解释 widgets 和 aggregators。通过 widgets, aggregators 和 interpreters 提供了一种上下文组件的编程方式来开发上下文感知应用。上下文感知分为对外界环境的感知和对内部资源的感知。

3) 自适应执行体。

自适应执行体用于动态调整实时任务,使之适应当前系统资源状况,并保持系统收益最优化,包括本地系统自适应和分布式系统自适应。实现动态配置和自适应的执行者,它能够在自适应策略的指导下,自动地将组件分配和组合。

4) 自适应机制。

降低系统中应用程序开发的复杂性,并使系统能在动态、不确定的环境下实现自适应资源管理。它是实现动态配置的前提,在确保系统可调度性的前提下使系统总收益最大化。

5) Sensor。

支持底层的传感器输入,接受来自外部环境信息的变化,同时占用较少的计算资源以及存贮资源。系统提供对传感器的封装机制。

●工作简介:

(1) 执行前的自适应准备。

通过上下文感知状态的变化和中间件内部构件的接口信息、资源使用情况的变化,自主决策推理,得到最优的“自适应动作执行序列”。

(2) 执行中的自适应动作。

根据执行前的自适应准备得到的自适应执行动作序列,采用优化的策略串行或并行执行相关动作。必须保证在执行的过程中,不破坏构件间的依赖关系、不破坏构件联合执行的完整一致性、不破坏核心的关注分离构件,并且满足资源的约束条件等。

(3) 执行后的自适应状态(两种状态)。

a. 过渡期:

在执行了自适应动作之后,处于过渡状态,如效果不好,仍可以逆向执行刚才的自适应动作,使之回到自适应之前的状态。

b. 无过渡期:

在执行了自适应动作之后,处于新的状态,不能回到自适应之前的状态。

7 结束语

从普适计算的特点出发,分析了普适计算环境下自适应技术的有关问题和需求,提出了构建基于普适环境的自适应系统的设计原则,探讨了自适应技术的实现方法,分析了具有自适应处理策略的目标任务在所需要的资源发生变化时,能够获得较为稳定的系统性能。目前,普适计算整个研究领域都还处于初期阶段,在原理和实践之间仍有一定距离,由于普适计算应用的特殊性,因此自适应技术在研究上的挑战性依然存在。

参考文献:

- [1] 徐光祐,史元春,谢伟凯. 普适计算[J]. 计算机学报,2003(9):1042-1050.
- [2] 刘 琳. 普适计算面临的挑战[J]. 重庆职业技术学院学报,2007,16(6):150-152.
- [3] 刘晓红. 浅析普适计算的现状与发展[J]. 广西医科大学学报,2008,25:99-100.
- [4] 魏 东. 普适计算的应用和展望[J]. 辽宁科技学院学报,2006,18(4):20-21.
- [5] 欧阳建权,王怀民,史殿习. 普适计算的可信研究[J]. 计算机应用研究,2008,25(12):3521-3524.
- [6] Tsyppkin Y Z. Adaptation and Learning in Automatic Systems [M]. Orlando, FL, USA: Academic Press, Inc., 1991.
- [7] Tekinerdogan B, Aksit M. Adapatability in Object - Oriented Software Development[C]// Workshop Report, In 10th European Conference on Object - Orented Programming. [s. l.]: [s. n.], 1996.
- [8] Preizy P, Oreizy O, Medvidovic N, et al. Architecture - based Runtime Software Evolution[C]// In Proceedings of the International Conference on Software Engineering. [s. l.]: [s. n.], 1998.
- [9] Yeo Bong Yoon, Hee Yong Youn. A New Service Discovery Scheme Adapting to User Behavior for Ubiquitous Computing [M]. Berlin Heidelberg: Springer - Verlag, 2005: 19-28.
- [10] 吴 卿. 面向普适计算的自适应中间件模型与方法研究[D]. 杭州:浙江大学, 2006.
- [11] 李 允,熊光泽,罗 蕾,等. 普及计算终端的自适应性技术研究[J]. 电子学报,2002,30(8):1121-1126.
- [12] 郭延辉,李蜀瑜,梁艳华. 面向普适计算的自适应中间件研究[J]. 贵州大学学报,2008,11(6):161-165.

第七届 CCF 中国计算机大会

(2010 CCF China National Computer Conference, CCF CNCC2010)

将于 2010 年 10 月 16—17 日在杭州隆重举行。