

Internet 环境下 分布嵌入式系统研究

—— 黄广君 著 ——



科学出版社

www.sciencep.com

TP360.21/24

2008

Internet 环境下 分布嵌入式系统研究

黄广君 著

本书由河南科技大学学术著作出版基金资助出版

科学出版社

北 京

内 容 简 介

嵌入式计算机系统以分布式系统方式加入互联网,是增强其功能、扩展其应用领域的重要途径。本书重点介绍构建开放的、跨平台的、具有实时保障机制的分布嵌入式系统所涉及的关键技术及最新研究成果,内容包括服务描述与服务发现、分布嵌入式系统架构、分布式中间件和实时任务调度算法。

本书以传统服务发布和服务引用模式为基础,将语义 Web 技术引入到分布式系统下的服务描述和服务发现中,实现服务功能的自动调用;阐述了基于本体的服务属性描述方法和基于逻辑推理与相似度计算的属性匹配技术;介绍了基于 Java 技术的系统实现方案,重点介绍了作为系统中中间件的嵌入式 Java 虚拟机的设计与优化技术,提出了一种基于预处理技术的动态自适应编译执行方式;针对嵌入式系统通常应用于实时过程的特点,在全面介绍了各种调度模型的基础上,重点讨论了两阶段调度策略、基于阈值的半抢占式调度算法,以及过载条件下多维优先级驱动高度算法等最新研究成果。

本书结构合理,适宜于从事嵌入式应用系统、分布式系统和实时过程领域的科学技术研究人员阅读,也可作为计算机软件、人工智能和计算机应用等专业高校师生的教学参考书。

图书在版编目(CIP)数据

Internet 环境下分布嵌入式系统研究/黄广君著. —北京:科学出版社, 2008

ISBN 978-7-03-020857-6

I. I… II. 黄… III. 微型计算机—系统设计 IV. TP360.21

中国版本图书馆 CIP 数据核字 (2008) 第 003744 号

责任编辑:耿建业 王志欣 / 责任校对:陈玉凤

责任印制:刘士平 / 封面设计:耕者设计工作室

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>

新蕾印刷厂印刷

科学出版社发行 各地新华书店经销

· *

2008 年 2 月第 一 版 开本: B5 (720×1000)

2008 年 2 月第一次印刷 印张: 10

印数: 1—3 000 字数: 192 000

定价: 32.00 元

(如有印装质量问题,我社负责调换〈环伟〉)

目 录

第 1 章 绪论	1
1.1 分布式系统的发展	1
1.1.1 分布式系统的产生	1
1.1.2 分布式计算技术的发展	2
1.1.3 分布嵌入式系统的特点	3
1.2 分布嵌入式系统研究现状	4
1.2.1 系统构建技术	4
1.2.2 服务描述与服务发现	6
1.2.3 服务质量 QoS 保障技术	7
1.2.4 当前存在的问题、本书解决方法及研究基础	8
1.3 本书研究内容及创新之处	9
1.4 本书组织结构	10
第 2 章 分布嵌入式系统设计技术	12
2.1 服务描述与服务查找	12
2.1.1 服务查找方式	12
2.1.2 基于本体的服务描述方式	12
2.1.3 基于描述逻辑的服务匹配	15
2.1.4 基于语义相似度计算的服务匹配	15
2.2 中间件与 KVM	16
2.3 KVM 设计技术	18
2.3.1 JVM 简介	18
2.3.2 KVM 的关键技术	20
2.4 QoS 保障技术	23
2.4.1 实时性调度算法概述	24
2.4.2 调度模型与可行性调度分析	28
2.4.3 多线程技术	30
2.4.4 线程池机制	31
2.5 本章小结	32
第 3 章 传统 Web Services 体系结构	33
3.1 设计目标	33

3.2 协议与标准	34
3.2.1 Web 服务描述语言 WSDL	34
3.2.2 统一描述、发现和集成协议 UDDI	38
3.2.3 简单对象访问协议 SOAP	47
3.3 分布式系统模型	53
3.3.1 总体结构	53
3.3.2 嵌入式设备系统层次结构	55
3.3.3 服务中心层次结构	56
3.3.4 QoS 层次结构	61
3.4 系统设计	62
3.4.1 发现模式	63
3.4.2 服务描述	67
3.4.3 工作模型	71
3.5 本章小结	74
第 4 章 KVM 设计	75
4.1 KVM 的特点	75
4.2 KVM 工作环境与内部结构	76
4.2.1 KVM 工作环境	76
4.2.2 KVM 内部结构	76
4.2.3 KVM 组成	77
4.2.4 类文件格式与存储分配	80
4.3 类的载入与建立过程	80
4.3.1 装载	81
4.3.2 连接	81
4.3.3 初始化	84
4.4 KVM 执行	85
4.4.1 本系统采用的执行方式	85
4.4.2 三种执行方式的性能仿真	90
4.5 KVM 安全性分析	91
4.5.1 Internet 环境安全性	91
4.5.2 建于 Java 语言和 VM 中的安全特性	92
4.6 类文件校验器	93
4.6.1 校验原理	93
4.6.2 预校验算法	94
4.6.3 KVM 在线校验算法	95

4.7 本章小结	95
第5章 KVM 中的性能优化及垃圾自动回收	96
5.1 预处理器设计	96
5.1.1 预处理器工作机制	96
5.1.2 预取算法与实现技术	97
5.1.3 本系统采用的预取技术	100
5.2 预处理子程序	100
5.2.1 划分基本块与循环	101
5.2.2 信息采集	102
5.2.3 大对象的预置	104
5.3 代码优化	106
5.3.1 对解释器的优化	106
5.3.2 对 JIT 的优化	106
5.4 内存管理与垃圾自动回收	110
5.4.1 KVM 中的内存管理	110
5.4.2 垃圾回收	112
5.5 本章小结	114
第6章 实时性实现方案	115
6.1 端到端时延模型	115
6.2 实时性调度算法	116
6.2.1 服务中心端的调度算法	116
6.2.2 服务方的调度算法	117
6.3 实时通信协议	124
6.3.1 基于交换机以太网实时性分析	126
6.3.2 总线结构以太网实时性分析	127
6.4 实时性垃圾回收算法	128
6.4.1 回收过程	129
6.4.2 回收算法的实时性分析	130
6.5 本章小结	131
第7章 一个原型实现	132
7.1 原型结构设计	132
7.1.1 需求分析	134
7.1.2 原型总体结构设计	134
7.1.3 原型模块功能设计	135
7.2 实验平台搭建	135

7.2.1 构建本体库	135
7.2.2 搭建注册中心	137
7.3 原型实现技术	138
7.3.1 服务描述机制	138
7.3.2 服务匹配算法	139
7.3.3 推理与映射	140
7.4 实例演示	141
7.4.1 具体服务描述与注册	141
7.4.2 具体服务查找	145
7.5 本章小结	146
参考文献	147
致谢	152

第1章 绪 论

分布嵌入式系统是指由嵌入式系统通过网络设施组成的分布式系统。嵌入式系统通常是指以应用为中心,以计算机技术为基础,软硬件可裁减,满足应用系统对功能、可靠性、成本、体积和功耗要求的专用计算机系统。嵌入式系统已广泛应用于家电、汽车、航空航天工程、工业设备和科学仪器。嵌入式计算机系统在应用数量上远远超过了各种通用计算机,它们具有一些共同特点,例如 CPU 计算能力较弱,内存容量较小和用户界面不完整等。根据嵌入式系统的内存容量,可以将其分为高端和低端两类产品。高端产品的开发环境和开发过程与普通的桌面计算机系统区别不大,一些运行在 PC 机上的应用程序可以便利地移植到高端嵌入式系统中。而低端产品则与桌面 PC 有较大区别,操作系统(若有的话)和应用软件的规模、内存消耗、算法复杂度和通信协议都有严格要求。由于高端嵌入式系统与普通桌面计算机差别不大,故本书将主要研究适用于低端嵌入式系统的分布式系统的一些关键技术。

目前,关于分布式系统还没有统一的定义。但一般可以描述为:一个分布式系统是一些独立的计算机的集合,但是对这个系统的用户来说,系统就像一台计算机一样。分布式系统所具有的这一性质称之为“单系统映射”^[1],这也是分布式系统与网络系统的根本区别所在。随着嵌入式系统应用技术和分布式系统构建技术的不断提高,特别是 Internet 应用的迅速扩展和深入,嵌入式系统越来越多地工作在分布式系统环境中。分布嵌入式系统应用可大致分为两大类:一是信息处理系统,如家庭网络 and 智能小区;二是过程控制系统,如工业监控、航空航天工程和国防工程等。这里主要以后一种应用为背景开展研究。

1.1 分布式系统的发展

分布嵌入式系统是分布式计算技术的一个新兴分支。在讨论分布嵌入式系统之前有必要了解一下分布式系统的发展历程。

1.1.1 分布式系统的产生

长期以来,控制系统的基本模式是由一台高性能的上位机作为控制中心负责监测、处理若干客户终端发出的服务请求。控制中心集中了几乎所有的计算资源,客户终端仅仅负责采集信息,相互之间没有直接联系。随着应用系统的不断扩大,

这种集中应用模式逐渐显露出弊端。首先,是控制中心性能的提高无法满足日益复杂的应用需求,依靠系统升级来提高性能的方式在经济上和技术上都是困难的;其次,系统的可靠性与安全性完全依赖控制中心,在规模与范围不断扩大的现实应用中这种做法是危险的;最后,移动网络等新的应用领域的不断产生使集中控制更加困难。于是,分布式计算技术便作为一种新的应用模式产生和发展起来。

从应用目的来看,分布式计算技术可以分为以高性能计算为目的的并行计算系统和以共享资源为目的的分布式系统。在不加特别说明的情况下,今后本书中分布式计算技术专指分布式系统。分布式系统按照应用模式又可分为客户机/服务器模式、纯分散模式(如 P2P 模式)和混合模式。客户机/服务器模式简单、可靠、性能稳定,但其可扩展性较低,主要用于小型应用系统。纯分散系统被认为是最标准的分布式应用模式,其良好的可扩展性和可靠性尤其适合构建大型应用系统,其代表性技术是基于 Web Services 的新一代 Internet 应用系统。但纯分散系统对各个单机系统具有较高的资源要求,并且缺乏对服务质量的控制机制,现实应用中大多采用混合模式,在系统中设置若干服务中心负责各个节点的接入控制、服务发现和实时调度。由于服务中心之间在信息管理上互为冗余,且计算资源分布在系统中各个节点,因此混合模式可以保持分布式计算的高性能与高可靠性特征。

1.1.2 分布式计算技术的发展

从 20 世纪 70 年代以来,对分布式系统的研究可以分为分布式操作系统和分布计算环境两大阶段^[2~4]。分布式操作系统力图直接管理整个系统中的各个计算机硬件,它在每个处理机上均运行一个相同的微内核副本,使各个计算机具有基本相同的硬件平台。典型的分布式操作系统有 Mach、Chorus 等^[5,6]。

由于在分布式系统中,每个节点必须安装相同的分布式操作系统,而现实中又没有得到一个得到广泛认可的分布式操作系统产品,因此在分布式操作系统所支持的分布式应用中仍旧缺乏足够的开放性和可扩展性。于是对分布计算技术的研究逐步转向中立于语言和操作系统的分布计算环境上来。

分布计算环境是在各节点不同单机操作系统之上提供一种透明的互操作能力。其早期典型代表是 OSF 的 DCE(distributed computing environment)^[5]。这是一个基于远程过程调用 PRC 的客户/服务器工作环境。DCE 满足了分布式应用的基本要求,但使用较复杂。随着面向对象技术的成熟,分布式对象计算技术逐步成为分布式系统的主流,其代表产品有 OMG 的 CORBA、Microsoft 的 DCOM、Sun 的 JINI^[7~11]。

在有 n 个节点的分布对象计算环境中,各节点的硬件平台或操作系统可能差别很大。为了保证任意两节点之间的互操作性,每个节点上应配置 $n-1$ 组引用接口和相应的实现对象。这将显著提高节点配置的代价和复杂程度,增大了软件

开发的工作量。因此,当前主流分布式系统普遍采用中间件技术^[7~12],即系统软件和应用软件之间,加入一个软件中间层,该层可以屏蔽各节点之间的平台异构性,向应用层提供一个统一的调用接口。这对于提高软件开发效率和降低维护成本具有重要意义。

由嵌入式系统构成的分布式系统在体系结构、任务调度、资源调用和网络协议等方面与桌面分布式系统有很大不同,但目的仍然是为了达到透明地共享资源、相互操作和提高可靠性。

1.1.3 分布嵌入式系统的特点

与普通分布式系统相比,分布嵌入式系统在构造技术和应用环境上具有一些显著特点。

1. 系统资源有限

由于嵌入式系统处理器资源和存储资源配置较低,应避免采用复杂的系统结构和高复杂度的求解算法。在桌面分布式系统中广泛使用的进程迁移与服务组合等技术应避免使用。分布嵌入式系统中的任务应尽可能功能单一,服务对象之间应尽可能松散耦合。分布嵌入式系统还应避免使用对内存配置与CPU计算能力要求较高的算法与数据结构,但简化算法不能通过简单地降低原算法的精度或完整性来达到简化的目的,必须利用分布式系统协同计算的能力并结合嵌入式系统的专用性特点来实现对算法的简化。简化可以包括三个方面:一是通过分布式协同计算预先或并行地对任务进行预处理,从而降低嵌入式系统在线工作负载;二是利用预处理所产生的信息进行适当的代码优化以提高系统性能;三是根据应用目的与环境的专用性裁减不会发生或发生概率很低的情形所对应的模块,从而缩小系统本身规模。

2. 各嵌入式设备的异构性

由于嵌入式系统可采用的微处理器与操作系统种类繁多,差异较大,在构建分布式系统时,通常采用中间件技术来屏蔽各个嵌入式系统的异构性,实现系统的相互操作和开放性,并提高开发效率。嵌入式系统所采用的中间件应具有高效精干特征,其本身尺寸及运行空间应尽可能小。

3. 应用要求的实时性

分布嵌入式系统应用通常对服务质量 QoS(quality of service)有严格要求,如端到端的优先级保留、调用延迟、抖动上限和一定带宽的预留等。这就要求分布嵌入式系统向用户提供带有时间约束的引用接口,并通过一定机制将用户的实时性要求映射到系统体系结构的各个层次。当应用对任务的端到端时延具有实时性要求时,系统必须在任务调度、消息传递和代码执行等环节提供实时支持。

1.2 分布嵌入式系统研究现状

分布嵌入式系统应用的日益广泛已经引起一些业内研究机构和著名公司的兴趣。目前对于分布嵌入式系统的研究主要集中在系统组织构建、服务描述与发现,以及服务质量 QoS 保障技术等方面,已产生了一些代表性系统。

1.2.1 系统构建技术

分布嵌入式系统的构建主要基于两类模式,即专用系统模式与通用系统模式。

1.2.1.1 专用系统模式

该模式从嵌入式设备出发,自下而上构建分布式系统。要求各设备具有相同的硬件接口,挂接在特定规范的总线上,组成一个高速的总线型分布式系统。该系统结构简单,组建方便,但对硬件有一定要求。通常专用于某个领域,如家庭影院、汽车电器等。具有一定影响的专用系统有 CEBus、EMIT 和 CAN 等。

1. CEBus(consume electron bus,消费电子总线)

CEBus 是微软领导的一个串行总线标准,该标准已被美国 ANSI(American National Standards Institute)接受,编号为 EIA-600^[13,14]。EIA-600 包括三部分:网络介质的物理设计与拓扑结构、数据协议、通用控制语言。CEBus 可以采用 7 种传输介质,其中主要采用 120V 动力线为传输介质,以同一速率(8000bits/s)传输数据。CEBus 支持嵌入设备的“即插即用”连接,但在 CEBus 中没有有一个控制中心。

2. EMIT

EMIT 是美国 Em Ware 公司开发的嵌入式微型网络技术,该技术包括 EM 微处理器、EM 网关和 EM 浏览器等部件,主要用来将各种 8 位或 16 位嵌入式系统连入 Internet 中^[14]。EMIT 为低端嵌入式系统提供了一个加入 Internet 的途径,与单机直接连入 Internet 技术相比,该方法具有灵活性和经济性等优点。

3. CAN(controller area network,控制域网络)

CAN 是由德国奔驰公司提出的一个串行数据通信总线,最初用于汽车内部多个控制器与传感器之间的数据交换,后来被国际标准化组织 ISO 所接受,编号为 ISO11898^[15,16]。CAN 总线协议本身只定义了物理层和数据链路层,其高层协议留给各厂家自行开发。CAN 工作原理与以太网有相似之处,同时它还通过“优先级逐位仲裁技术”为数据交换提供实时性支持。CAN 曾广泛用于工业现场总线,目前有被基于以太网技术的分布式自动化接口 IDA 取代的趋势。

除上述系统外,国内许多研究机构也对分布嵌入式系统的体系结构、调度算法和实现技术进行了研究。浙江大学计算机系统工程研究所等单位联合开发的家庭

网络系统为智能家电的相互操作提供了一个网络平台;清华大学、上海交大、东南大学、东北大学、哈尔滨工业大学等单位也对现场总线技术和嵌入式 Internet 技术进行了研究。

上述系统尽管在各自领域获得较大成功,但其特殊的接口或线路要求限制了它们的开放性、扩展性和应用的广泛性。因此,目前关于系统构建技术的研究主要集中在通用分布式系统模式。

1.2.1.2 通用系统模式

通用系统采用中间件技术屏蔽各设备平台的异构性并提供统一的编程接口,从而具有良好的扩展性和开放性。通用系统的代表技术是 CORBA 和 JINI。

1. CORBA

这是 OMG(object management group)发表的一个分布式系统技术规范。CORBA 被认为是最完整的分布式系统规范,它具有良好的开放性和跨平台特性,能够最大限度地屏蔽各设备的异构性,实现各设备之间的相互操作。但在 CORBA 实现中,系统层次较为复杂,各个部件的实现和配置要求较严,工作效率较低。因此,通用 CORBA 系统不能直接用于实时环境和嵌入式环境。

为了适应嵌入式系统应用的要求,OMG 发表了一个专用于分布嵌入式系统的技术标准 minimum CORBA^[17]。它是在标准 CORBA 的基础上,通过裁减 CORBA 的动态机制来减小系统尺寸,使其适用于嵌入式环境。minimum CORBA 裁减动态机制的依据是嵌入式系统的专用性,即嵌入式系统功能有限,其访问的其他站点也是有限的。在通常情况下,这一裁减不会影响 minimum CORBA 的使用,但在某些情况下,可能影响 minimum CORBA 的可扩展性。

minimum CORBA 在高端嵌入式系统中有一定的市场,但它还不能在低端嵌入式系统中得到广泛应用,主要是因为 minimum CORBA 要求各站点至少要有 1Mbytes 的内存空间,而低端嵌入式系统通常只有 512Kbytes 或 256Kbytes。minimum CORBA 层次结构的复杂性也在一定程度上限制了它在低端嵌入式设备中的应用。此外,minimum CORBA 没有提供实时性应用机制也限制了其应用范围。

在 CORBA 系列中,实时性应用机制由 Real-time CORBA 规范来定义^[18]。该规范定义了一组标准接口和策略以供用户来配置和控制系统对其资源、内存资源和通信资源进行处理,为实时性应用提供端到端时延的可预测性。该规范没有对设计和实现技术作出规定,因而出现了多种风格的 CORBA 系统。该规范的典型实现是 Washington 大学计算机系的 TAO(the ACE ORB)系统和 Rode Island 大学计算机系的 NraD/URI CORBA^[19,20]。

Real-time CORBA 规范虽然提供了一个较完整的实时分布式系统架构,但其已有版本是针对桌面系统定义的,关于嵌入式环境的实时规范目前还在定义之中。

因此在 CORBA 系列中,目前还没有一种完整的适用于实时嵌入式应用环境的技术规范。

2. JINI

这是由 Sun 公司发布的基于 Java 的分布嵌入式系统体系规范。JINI 的目标是以 Java 虚拟机 JVM(Java virtual machine)为中间件构筑一个动态自管理的网络平台。包括硬件设备和软件程序在内的各种资源可以简便地、透明地相互共享。JINI 技术的核心是通过 JVM 这一中间件将各个异构系统映射为同构系统,从而使各设备都能以 Java 接口方式进行交互。

JINI 系统由一个或若干群体组成,每个群体中有一个或若干个高性能 PC 或高端嵌入式系统作为服务中心,向其他设备提供服务发现和租约管理等服务功能。各节点之间通过若干标准接口实现相互操作。与 CORBA 相比,JINI 既有集中控制的机制,又具有结构松散简单的特点,这对于嵌入式环境的应用非常重要。

尽管 JINI 是专为嵌入式系统环境发布的技术规范,并已得到广泛应用,但 JINI 技术规范也有一些先天不足。第一,由于 JINI 发布时间较早,它采用的 Java 虚拟机是通用 JVM 版,而不是后来专为嵌入式设备发布的 KVM 版,系统对内存需求较大。第二,在 JINI 系统中,JVM 处于单机工作状态,只能利用所在平台的处理器及内存资源完成 Java 应用程序的检验和执行。因此,JVM 往往成为 JINI 系统的工作瓶颈,限制了 JINI 系统在低端嵌入式设备中的推广应用。第三,JINI 没有提供任何接口和机制来反映嵌入式设备对时间特性的要求,也没有在群体一级提供安全管理和 Internet 环境下的安全协议网关。这使得 JINI 很难用于构建 Internet 环境下的实时系统。

1.2.2 服务描述与服务发现

分布嵌入式系统的单一映像特性要求各个节点设备之间的服务引用必须是位置透明和名字透明。这种透明性要求系统提供一种高效和实时的服务发现机制。服务发现包括查找与匹配两个方面。服务查找方式与分布式系统的组织结构紧密相关,例如在 CORBA 中,客户请求的服务对象经过 ORB 中间件传递给服务器方的对象适配器,该适配器负责定位提供服务的对象实现。而在 JINI 中,服务查找完全由查找中心代理完成。目前对于服务查找的研究主要集中在查找范围的可扩展性与查找活动的并行性上。

服务匹配技术是服务发现的基础,也是当前的一个研究热点,研究人员已提出三种服务匹配技术:一是基于关键词的匹配查找方式,该技术查全率较高但准确率较低;二是基于框架的匹配查找方式,具有较高的准确率,但查全率较低;三是演绎查找方式,依靠服务的逻辑表示与逻辑推理来判定请求的服务与目标服务是否匹配,该方式目前还不能应用到实际系统中^[21,22]。总的看来服务匹配技术的研究还

有待新的突破。

与匹配技术直接相关的一项技术是服务描述。过去对于服务描述的研究大多集中在词法和句法层次上,人们试图建立一个通用的描述框架,配以完整的数据辞典,从而实现对各种服务的无歧义描述。但到目前为止,除了在一些小型专用系统中取得进展之外,还没有发展出得到广泛认可的服务描述规范。随着 Internet 应用的不断扩展,特别是 Web Services 技术的兴起,基于专有领域知识的本体论(ontology)成为研究热点^[23]。人们开始探索在服务描述中增加语义成分的有效技术,并取得了一些进展^[22]。

1.2.3 服务质量 QoS 保障技术

随着分布嵌入式系统应用水平的不断提高,人们对它为应用所提供的服务质量 QoS 的要求也在提高。与网络中对 QoS 的要求不同,分布式系统不仅在数据传输环节中的有 QoS 指标要求,而且对设备处理能力、服务请求接入能力和系统对任务的调度能力都有完整的要求。

目前还没有一种分布式系统能够提供完整的 QoS 保障机制。虽然实时 COR-BA 等系统可以提供任务级的端到端时延预测,但对于整个任务集的调度处理还无法满足实时要求。当前人们对 QoS 的研究主要集中在网络传输性能上。其代表性技术是 ATM 论坛的 ATM QoS 机制、IETF 制定的 QoS 机制以及 IEEE 小组提出的 IEEE802.1d^[24~27]。其中 ATM 技术具有最完善的网络 QoS 保障体制,可以胜任各种传输业务的要求,但由于技术复杂和价格昂贵,很难应用于嵌入式环境。IETF 制定的 QoS 机制包括集成服务 Intserv、差分服务 Diffserv、QoS 路由和多协议标记交换 MPLS 等技术,主要针对 Internet 多媒体传输业务。IEEE802.1d 基于优先级将任务划分为不同队列,为局域网的实时数据传输提供了一定能力,但 IEEE802.1d 及后续的 IEEE802.1w 在端口转换过程中延迟较大,其应用范围难以广泛^[27,28]。

与网络通信中的 QoS 研究相比,针对分布嵌入式系统调度算法的研究还很不足。分布嵌入式系统所采用的调度方法和理论可以看作是对单处理器条件下基于单优先级的周期性任务调度算法的扩充。由于应用目的和应用环境不同,分布嵌入式系统所采用的调度算法与普通分布式系统有很大差异。在分布嵌入式系统中,通常不需要将一个任务进程分解为 m 个子进程(或线程),再均衡地分配到 n 个处理机上去执行。分布嵌入式系统关心的是在某个时刻它能接纳多少服务请求,以及如何保证这些请求能够在要求的时间内得到响应和处理。在任务调度中,应力求以较小的调度开销取得较高的调度成功率。由于嵌入式系统难以掌握其他节点设备的工作状况,因此设计人员倾向于在系统中设置若干调度中心,负责对服务请求的接入控制。

一个完整的实时保障机制必须考虑各个节点设备代码执行的实时性。经典调度理论和大多数调度算法都假定任务具有一个预计执行时间(即无干扰条件下的执行时间) C_i ,并依据 C_i 进行可调度分析。而实际应用中,嵌入式系统的负载状况、操作系统版本、编程方式和采用的算法都对代码的执行发生着影响。 C_i 的实际大小存在一个抖动值,有时该抖动值会达到影响实时调度的程度,因此代码的实时执行问题也是实时分布式系统的一个研究方向。目前研究人员的解决方法是将求解问题适当分解后,利用多线程同步技术予以实现。

1.2.4 当前存在的问题、本书解决方法及研究基础

综上所述,分布计算技术在系统结构、服务发现和实时性保障等关键技术上的研究还处于发展之中,而分布嵌入式系统在资源配置和应用环境上的固有限制增加了系统设计和实现的难度。因此,目前的几种主流分布式系统都难以满足日益广泛的分布嵌入式应用环境对于开放的、可扩展的、可靠的和实时性的计算平台的迫切需要。为了解决这一问题,本书将根据分布嵌入式系统的上述特点,设计一种适用于低端嵌入式系统的、可扩展的、具有实时性保障机制的分布嵌入式系统体系结构。

在设计中以局域网为基础构建一个分布式系统,称为一个群体,每个群体对应一个家庭网络、小型公司、生产车间、飞行器系统或军事控制系统。群体之间可以通过 Internet 设施相互结合成为更大范围的分布式系统联合体。

根据嵌入式系统特点,可以采用混合发现模式进行服务查找。在每个群体中设置一个或若干个服务中心负责设备管理、接入控制、服务查找与转发、任务调度,以及远程文件的缓存和预处理。与集中式系统中的主机或服务器不同,服务中心并不对客户服务请求直接进行处理与应答,它只是提供某种管理与帮助。系统的计算资源仍然分布在各个嵌入式设备上。各个服务中心之间以及联系密切的群体之间保持一定程度的冗余备份以提高系统的可靠性。在分布式系统中,各个嵌入式设备既可以是客户方,也可以做服务方。利用服务中心的帮助,各个嵌入式系统设备可以即插即用,设备之间能够透明地相互操作、共享资源。

为了屏蔽众多嵌入式系统平台的异构性,采用 Java 技术构建分布式体系结构,并采用 J2ME(Java 2 micro version)的虚拟机 KVM(Kilos virtual machine)作为中间件为各个嵌入式系统提供统一的编程和执行环境。J2ME 是 Sun 公司专为嵌入式应用环境发布的一个 Java 版本。之所以选择 J2ME,是考虑到它的跨平台特性、安全性和开发环境便利齐全等优点。针对 Java 程序执行速度较低这一缺点,本书提出了基于预处理技术的 KVM 设计思想,在服务中心配置一个多进程类文件预处理器为各个设备中的 KVM 进程提供类文件的远程预取、字节码校验和数据流分析,以减少 KVM 的工作负担并提高其执行速度。由于预处理器与 KVM

之间可以相互协同、并行工作,因此预处理器的设置可以在很大程度上掩藏类文件下载过程中 Internet 线路所产生的延迟。此外,利用预处理得到的有关信息,KVM 在代码优化和无用单元回收等环节获得帮助,从而更大程度地提高代码执行速度和降低对嵌入式系统的配置要求。由于 KVM 是构建本系统的关键部件,本书将把它作为重点之一进行讨论。

本系统在应用层为客户提供一组包含时间约束参数的服务引用接口。提出服务请求的客户设备可以通过时间参数描述对于实时性的要求,该要求通过层次结构逐级映射到任务调度、消息传递和代码执行的环节。通过集成有关机制,系统对任务提供端到端实时性支持。根据分布嵌入式系统中任务相对独立简单的特点,在任务调度中采用二级调度策略简化调度算法并控制任务接入;通过多维优先级调度算法解决过载调度问题提高系统的健壮性;通过一种基于回溯的有限抢占调度算法降低任务切换开销并提高调度成功率。

根据嵌入式系统应用目的通常专一的特点,在服务描述、结构层次和通信协议等方面还提供了集成框架,为系统设计的灵活裁减与选择提供了方便。

本书的研究基础是某导弹地面仿真系统项目和河南省自然科学基金课题“基于中间件的分布嵌入式系统研究”。在课题研究中,基于 Java 技术构建分布式系统平台,通过开发一个 KVM 部件作为中间件,实现了系统的通用性和开放性。与单机环境下的 KVM 不同,该 KVM 与配置在服务中心上的类文件预处理器并行工作,这使得 KVM 可以实时地利用预处理所采集的信息对字节码进行校验和优化,从而提高系统性能。此外,通过在任务请求接口和服务调用接口加入时间属性,为系统引入实时性特征,并在任务调度、通信协议和 KVM 实现等环节提供实时性支持机制。由此构建的系统平台可以用于工业监控和国防工程等领域。

1.3 本书研究内容及创新之处

本书研究内容包括以下几个方面:

(1) 提出一个具有实时性保障机制的分布嵌入式系统体系结构,并基于 Java 技术设计了系统的主要框架和主要部件。在系统中,各节点设备通过带有时间属性和其他优先级属性的调用接口相互透明地访问资源。作为控制节点的服务中心为各节点提供基础服务设施,如服务发现、安全协议网关和 Java 类文件预处理等。

该系统采用一种基于并行查找的可扩展的服务发现模式,可以在有限的时间范围内判断一个服务请求能否在系统的一定范围内得到实时处理。作为服务发现的基础,本书还定义了一个带有语义信息的扩展属性描述树,树中节点可以携带度量约束规则和语义推理规则集。

(2) 全面讨论了 KVM 的设计及优化技术。设计了一个基于预处理技术的动

态自适应 KVM 系统,该 KVM 系统通过与服务中心并行工作,提高了 KVM 性能并掩藏了 Internet 线路所产生的较大延迟。该系统还利用预处理结果选定最有优化价值的热点代码段,并采用方法级和基本块级两级优化技术对其进行优化,从而提高 KVM 的执行速度。

(3) 在对任务端到端时延划分的基础上,研究了各个环节的实时保障问题。首先提出了一个两阶段调度策略及相应算法。根据嵌入式系统应用目的专一和各节点任务相对独立的特点,将分布式系统由多处理器多任务调度模式简化为两个阶段的单处理器多任务模式。第一阶段采用基于多维优先级的调度算法完成对任务集的全局调度、接入控制和过载处理;第二阶段采用基于回溯的有限抢占算法或过载调度算法对本地任务集进行调度。

(4) 提出了一种优先级驱动的实时以太网协议。该协议在传统实时调度理论基础上考虑了通信任务的延迟抢先特性和时间重叠性,在保证高优先级信息的低延时性同时,还保持了与非实时系统的兼容。

(5) 讨论了程序执行的实时问题,重点分析了对程序执行时间的确定性影响最大的无用代码回收问题。采用 Petri-net 技术讨论了回收过程的资源竞争与并行性处理技术,并给出了一个基于多线程并发处理的实现方案。

本书创新之处有以下几方面:

(1) 定义了一个基于本体的扩展属性描述树,并给出了相应的属性与属性集的匹配规则。

(2) 提出了一个基于预处理的 KVM 设计思想。利用分布式系统协同工作能力,由服务中心的预处理模块预先下载相关的类文件,并对其进行安全校验和采集信息。KVM 利用预处理得到的信息可以有效地提高代码执行速度,并降低对内存空间的配置要求。

(3) 提出了一个基于回溯的双优先级有限抢占式调度算法。与完全抢占式调度算法相比,该算法具有调度成功率高、切换开销适中等优点,同时该算法本身具有有限的时间开销。

(4) 提出了一个具有有限停滞时间的无用代码回收算法。KVM 中确保实时性的主要障碍在于无用代码回收的不确定性,本书采用多代计数回收算法和多线程同步机制,在不增加内存消耗的条件下,保证了回收算法的可实时调度性。

1.4 本书组织结构

全文共分七章,各章内容安排如下:

第 1 章介绍了分布式系统发展,分布嵌入式系统特点、研究现状,本书的解决方法和研究基础及研究内容和目标。