

DCAM 平台下的分布式方面构件模型及 协同框架的研究*

王 斌, 丁 建, 盛津芳

(中南大学 信息科学与工程学院 计算机科学与技术系, 长沙 410083)

摘 要: 由于方面化中间件平台 DCAM 的方面内核实现机制没有提供对分布式环境下方面构件通信的支持, 扩展了 DCAM 中方面内核的方面调用机制, 提出了分布式方面构件模型以及分布式环境下方面构件的协同框架。扩展后的 DCAM 平台不仅保持了原平台对 POJO 的支持, 同时通过对分布式协同框架的引入, 使平台在分布式环境下具有更好的可扩展性和容错性, 满足了按需计算的需求。最后通过应用实例对 DCAM 的分布式模型进行了可行性验证并给出了性能分析。

关键词: 双核中间件; 分布式方面中间件; 分布式方面构件模型; 协同框架

中图分类号: TP311.52 **文献标志码:** A **文章编号:** 1001-3695(2010)08-2959-03

doi: 10.3969/j.issn.1001-3695.2010.08.039

Research of distributed aspect-components model and collaboration framework for DCAM

WANG Bin, DING Jian, SHENG Jin-fang

(Dept. of Computer Science & Technology, School of Information Science & Engineering, Central South University, Changsha 410083, China)

Abstract: The core realization mechanism of distributed aspect middleware-DCAM doesn't support component communication in distributed environment. The paper firstly extended the calling mechanism of aspect core, and then proposed distributed aspect-components model and collaboration framework for DCAM. Not only did the extended DCAM keeps up supporting for POJO, but also it made the platform have better scalability, fault tolerance and satisfy the requirements of on-demand computing better in distributed environment through the introduction of the distributed collaborative framework. Finally, the paper validated the distributed model of DCAM with an application and analyzed its performance.

Key words: dual core middleware; distributed aspect middleware; distributed aspect-components model; collaboration framework

随着互联网的不断普及以及计算资源的多样化和差异化, 传统的中间件技术已经无法胜任普适计算、按需计算等一些新型计算模型的要求。面向方面软件开发 (aspect-oriented programming, AOP) 为上述计算模型中关键问题的解决提供了一种可行的方法^[1]。AOP 通过分离横切到构件中的方面来控制关注点的纠缠, 从而能够简化构件的组装, 降低系统的复杂性, 提高系统的自适应能力^[2]。目前在基于 AOP 的中间件理论和技术研究, 代表性研究成果有 JAC (Java aspect components)^[3,4]、DyMAC^[5,6] 和 Damon (distributed AOP middleware on top of a P2P overlay network)^[7,8]。JAC 通过 RMI 和 CORBA 底层通信机制实现了分布式横切关注点的编织与调用。DyMAC 解耦了构件与中间件的代码纠缠, 从而使中间件能够通过显式接口为通知构件提供良好的模块化能力。Damon 提出了方面化构件模型以及方面远程服务的概念, 并采用 P2P 协议作为分布式方面的通信承载协议, 以支持互联网上大规模分布式应用系统。这些中间件技术虽然提出了一些分布式 AOP 的实现机制, 然而却存在一些缺陷。JAC 和 DyMAC 不能根据需要扩展支持服务, 或者快速、便捷地部署第三方提供的服务功能, 同

时对网络的要求比较高, 在不稳定网络环境中无法正常工作。Damon 对分布式通信协议支持有限且平台扩展性不强。因而, 这些 AOP 中间件技术并不能够真正地满足分布式环境下按需计算和自适应计算的需求。

DCAM (dual-core aspectual middleware) 是本文提出的一种方面化中间件模型, 它具有较好的可扩展性、依赖关系管理能力和细粒度 AOP 支持^[9-11]。本文对 DCAM 平台的方面内核机制进行了扩展, 提出了分布式环境下的分布式方面构件模型及协同框架, 以支持分布式环境下按需计算和自适应计算的需求。

1 DCAM 平台分布式方面构件模型

方面编织模型主要研究方面编织的过程模型。编织模型决定了各种通知的实现方式, 同时决定了基于该编织模型的面向方面开发环境所支持的编程模型。DCAM 平台中方面编织模型是基于类修改和反射的动态方面编织模型 DAWeaver (dynamic aspect weaver)^[9,12,13]。为满足 DCAM 平台分布式环境下方面构件编织与调用的需求, 对 DCAM 平台的 DAWeaver 进行了扩展。扩展后的分布式动态方面编织模型如图 1 所示。

收稿日期: 2009-12-22; **修回日期:** 2010-02-27 **基金项目:** 国家自然科学基金资助项目(60970039); 湖南省自然科学基金资助项目(07JJ6124)

作者简介: 王斌(1973-), 男, 副教授, 博士, 主要研究方向为软件工程; 丁建(1983-), 男, 硕士, 主要研究方向为面向方面软件工程及方面编织; 盛津芳(1971-), 女, 副教授, 博士, 主要研究方向为基于构件的软件工程(jfsheng@csu.edu.cn)。

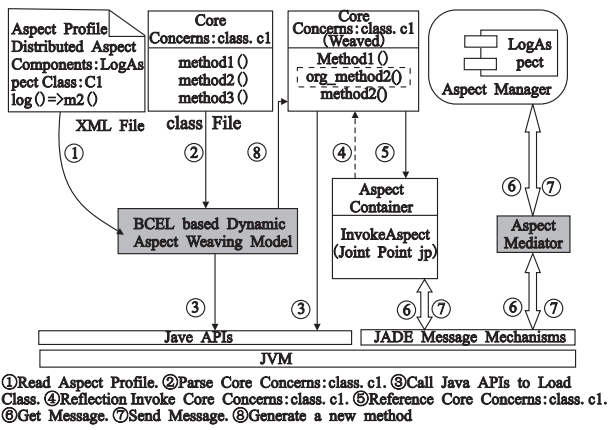


图1 分布式动态方面编织模型-DDAWeaver

图 1 中,方面配置文件(Aspect profile)采用 XML 语言定义系统关注点与核心关注点的耦合关系。核心关注点构件处理业务逻辑,分布式方面构件封装系统关注点的实现部分,两类构件可独立开发。方面容器(Aspect container)依据方面配置文件,调用方面构件。基于 BCEL 的方面编织内核依据方面配置文件分析核心关注点类,对需要加入方面构件的方法,如 method2 进行以下操作:a)重命名 method2 方法为 org_method2;b)生成新方法,方法名为 method2,其参数、返回类型以及访问权限与原方法相同;c)向新方法体中插入对方面容器中某特定方法的调用。

分布式方面构件的调用过程经过以下几步:a)获取目标组件上下文信息;b)方面容器向方面中介者(Aspect mediator)发出对指定方面构件的调用消息;c)方面中介者获取并存储调用信息,同时向目标构件发出调用信息;d)方面中介者获得并存取目标构件的反馈信息;e)方面容器通过方面中介者获取调用的反馈信息。

扩展后的分布式动态方面编织模型(distributed dynamic aspect weaver, DDAWeaver)对 DAWeaver 在三个方面进行了扩展:首先,采用消息传输机制实现分布式方面构件的调用;其次,提出了具有消息处理和行为能力的分布式方面构件模型;最后,将分布式方面构件的管理转移到方面管理器,而方面容器主要负责管理分布式方面构件的配置和调用信息。

同时,由于 DDAWeaver 并没有改变 DAWeaver 基于类修改和反射的动态方面编织机制,扩展的只是分布式方面构件调用的实现机制,DDAWeaver 不仅能够实现分布式方面构件的调用,同时还保持了对原方面构件模型和 POJO 的支持。原 DCAM 平台中并没有明确提出方面构件的模型、约束和规范等。为支持 DDAWeaver 下分布式方面构件的调用,本文引入构件通用规范,提出了分布式方面构件模型,如图 2 所示。

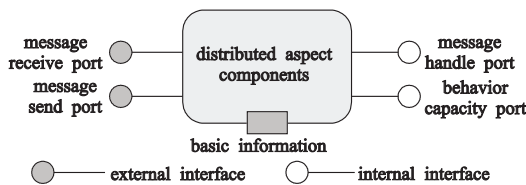


图2 分布式方面构件模型

在 DDAWeaver 中,方面容器与分布式方面构件之间采用消息传输机制实现方面的动态调用。因此分布式方面构件必须具备接收消息和消息处理的能力。分布式方面构件模型主要由以下五个部分构成:

a)消息接收端口。接收方面容器发来的调用请求消息,

并将请求消息放入分布式方面构件的消息队列中,同时触发相应的事件,通知分布式方面构件的处理方法及时处理方面调用的消息。

b)消息发送端口。将处理的结果返回到方面容器或进行存储,如以特定方式存于文件或数据库。

c)消息处理端口。从消息队列中获取待处理的消息,由行为能力处理端口对消息进行分析和处理。在消息处理端口中可以有多个行为能力端口来处理不同类型的消息。

d)行为能力端口。定义分布式方面构件所具有的行为能力,可以进行扩展。

e)构件基本信息。主要用于描述分布式方面构件的基本信息,这些信息包括分布式方面构件的统一标志符、状态信息以及所属的方面管理器。

与 Damon 提出的方面化构件模型相比,DCAM 的分布式方面构件模型具有很好的行为能力,可以把它视为一个智能体,从而为 DCAM 实现中间件平台的上下文感知和动态演化奠定了基础。

2 分布式方面构件协同框架 DACCF

分布式方面构件协同框架(distributed aspect-components collaboration framework, DACCF)的主要目的是实现分布式方面构件的分布式管理和调用,并为其提供必要的支持机制,如图 3 所示。

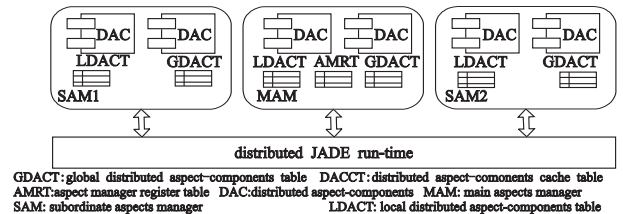


图3 分布式方面构件协同框架DACCF

在 DACCF 中,以方面管理器集群的方式对分布式方面构件进行管理。在 DCAM 平台中有且仅有一个主方面管理器,而可以有多个分布在同一域内不同节点的从属方面管理器。从属方面管理器负责管理自己节点的分布式方面构件,并为其提供运行环境,管理其生命周期和消息分发支持。同时,方面管理器提供了相应服务支持分布式方面构件的克隆以及在方面管理器之间的移动。在平台启动时,从属方面管理器的启动依赖于主方面管理器,它们只有在主方面管理器的方面管理器注册表(aspects manager register table, AMRT)中注册后才能启动。图 4 展示了 DACCF 中各元素的关系。

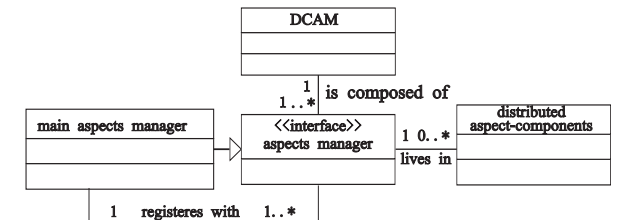


图4 DACCF元素关系图

从图 3 和 4 中可以看出,主方面管理器是整个 DACCF 的核心部件,其中最为重要的是 AMRT 和全局分布式方面构件表(global distributed aspect-components table, GDACT)。AMRT 负责管理当前 DCAM 平台的从属方面管理器,该表存储同一域内所有从属方面管理器节点的对象引用和地址信息。

GDACT 负责维护全局方面构件,它存储了当前 DCAM 平台中所有分布式方面构件的基本信息,如分布式方面构件的统一标志符、当前状态和所属的从属方面管理器。

下面重点介绍 DACCF 的分布式缓存机制、分布式实例控制机制、消息传输服务和事件通知服务的实现机制。

1) 分布式缓存机制 (distributed cache mechanism, DCM) GDACT 存储了当前 DCAM 平台下所有处于活动状态的分布式方面构件的信息,当方面容器通过该节点的从属方面管理器调用分布式方面构件时,如果直接搜索 GDACT 获得目标构件所在的从属方面管理器,那么将造成 DACCF 的性能瓶颈。为此,引入分布式方面构件缓存表 (distributed aspect-components cache table, DACCT) 存储与从属方面管理器最近通信的全局方面构件信息,而本地分布式方面构件表 (local distributed aspect-components table, LDACT) 则存储本地分布式方面构件的信息。这样,分布式方面构件的调用,一般并不涉及主方面管理器,而仅仅由 DACCT 和两个参与通信的从属方面管理器共同完成。对 GDACT 的搜索只有在搜索失败的情况下进行,并将搜索结果存储于 DACCT。

由于 DACCF 具有很好的动态性,它能够支持分布式方面构件的移动,以及动态部署与卸载,这样 DACCT 中可能会存储一些无效的分布式方面构件信息。在这种情况下,从属方面管理器将根据分布式方面构件行为及时更新 DACCT。为了提高 DACCT 中缓存数据的使用率,对 DACCT 的更新采取最近最少使用 (least recently used, LRU) 替换策略。

2) 分布式实例控制机制 (distributed instances control mechanism, DICM) 由于整个 DACCF 对主方面管理器具有很强的依赖性,一旦主方面管理器失效将影响整个框架的正常运行。为解决这一问题,在 DACCF 中提供了一个可配置的控制层来管理主方面管理器的分布式实例,实现对实例的分布式引导和分布式控制。同时,也将从属方面管理器列入克隆服务的考虑范围,作为控制层的一部分,从而使整个框架具有很好的容错性。

3) 消息传输服务 (message transport service, MTS) DACCF 采用基于消息传输的通信机制实现平台内部以及平台之间的信息交互。MTS 支持消息的同步和异步传输,它主要负责处理方面容器与方面管理器之间以及方面管理器之间的消息交互。

4) 事件通知服务 (event notification service, ENS) 它负责处理每个节点所有触发事件的分布式通知,在所有方面管理器中都有该服务。它能够对方面管理器的事件进行拦截,并通知订阅该事件的对象,进行相应的操作。通过 ENS 可以监视和跟踪平台内部元素的运行状况,如分布式方面构件和从属方面管理器的状态。

3 性能对比分析

为测试扩展后 DCAM 平台上编织后的目标系统的执行效率,本文选取购物车实例模仿购物车添加和删除商品等主要操作。在 DAWeaver 和 DDAWeaver 对目标系统在相同试验环境下选取如下相同测试用例:执行购物车 shoppingCartOperator 对象中的添加商品的 addShoppingCartItem 和移除商品的 removeShoppingCartItem 方法 n 次 (n 分别为 $10^2, 10^3, 10^4, 10^5$)。其中 addShoppingCartItem 方法已织入方面 logAspect.staticLog (该函数为空函数)。为减少误差,每次测试执行 50 次,取平均值。运行环境为: Intel® Pentium® Dual 处理器, 2.0 GB 内存, Win-

dows Server 2003 操作系统, Sun 1.6.0 JVM 虚拟机, Eclipse 3.2 开发环境。

图 5 给出了目标系统分别在 DAWeaver 和 DDAWeaver 下运行的性能对比结果。从图中可以看出,目标系统在 DDAWeaver 中的执行速度要慢于 DAWeaver,同时,随着重复次数的增加,性能差距逐渐加大。这种性能差距归因于两者调用方面构件的实现机制不同。DAWeaver 通过 aspect container 反射调用方面以及原方法,不存在网络延迟的影响,因而执行效率较高。DDAWeaver 由于分布式环境下的网络延迟和消息处理能力的影响,基于消息传输的分布式方面调用执行效率低于 DAWeaver。

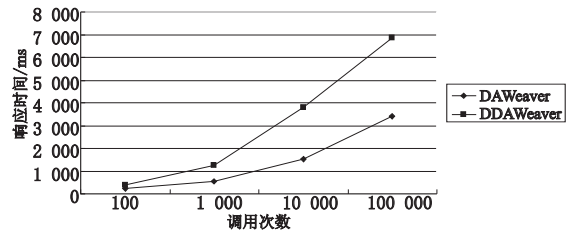


图5 性能分析对比图

DDAWeaver 虽然存在性能方面的损失,但是实现了分布式环境下分布式方面构件的调用。分布式方面构件是具有一定行为能力的智能体,为实现 DCAM 平台的上下文感知和动态演化奠定了基础。同时 DDAWeaver 符合按需计算中间件平台需求,可依据用户或数据传输的需要,动态组合计算模块,提高了系统的自适应能力。对于 DDAWeaver 性能上的损失,可以通过负载均衡机制来弥补。

4 结束语

分布式方面构件模型及协同框架的研究是面向方面自适应中间件的基础性研究工作。本文对原 DCAM 平台的 DAWeaver 进行了扩展,提出了分布式方面构件模型及分布式方面构件协同框架 DACCF,实现了分布式方面构件的动态编织与调用及分布式管理。同时,对比分析了由 DDAWeaver 编织的目标系统的执行性能。在上述研究基础上,未来将进一步完善分布式方面构件协同框架的相关实现机制,如扩展 DACCF 所支持的消息传输协议,优化分布式缓存机制和分布式实例控制机制的性能。

参考文献:

- [1] ROUVOY R, BEAUVOIS M, ELIASSEN F. Dynamic aspect weaving using a planning-based adaptation middleware [C]//Proc of the 2nd Workshop on Middleware-Application Interaction. New York: ACM Press, 2008: 31-36.
- [2] ROUVOY R, BEAUVOIS M, ELIASSEN F. Dynamic planning and weaving of dependability concerns for self-adaptive ubiquitous services [C]//Proc of the 2009 ACM Symposium on Applied Computing. New York: ACM Press, 2009: 1021-1028.
- [3] JAC home page [EB/OL]. <http://jac.ow2.org/>.
- [4] NAVARRO L D B, SUDHOLT M, VANDERPERREN W, et al. Explicitly distributed AOP using AWED [C]//Proc of AOSD 2006. New York: ACM Press, 2006: 51-62.
- [5] LAGAISSE B, JOOSEN W. True and transparent distributed composition of aspect-components [C]//Proc of Middleware'06. New York: Springer-Verlag, 2006: 42-61.

(d)复制新染色体到初始种群中,并转至步骤 b)。

d)输出代价值最小的染色体,解码出此相位状态并作为各路口相位控制信号。

在以上优化进化过程中,需要不断为系统产生新的车辆。为了保证系统的稳定性,在进化过程中,要求每次产生的车辆数量与分布规律相同。实际这样操作获得了比较好的效果。

3 仿真结果及分析

在以上模型的基础上,笔者开发了城市智能交通仿真优化系统。选择佛山市某城区交通地图为基础,建立了 72 个交通控制路口,设置了 8 条不同的线路(线路可随意设置),并对不同的路口设置不同的交通规则。同时为了简化模型,假定:a)仿真时所有车辆以相同的加速度启动与刹车;b)车辆行驶过程中不改变车道;c)各种车辆在同一车道上行驶速度相同(不允许超车);d)各车长度相同;e)GA 优化时,车辆产生采用均匀分布,演化仿真时采用泊松分布。采用改进遗传算法对其进行相位差优化,对比优化前后的仿真结果如表 2 所示。

表 2 优化前后的仿真结果对比

数据项	GA 优化前		GA 优化后		进化仿真条件
	平时时	高峰时	平时时	高峰时	
随机产生的车辆数量/辆	117	142	126	164	群体大小:50
每车平均通过路口数	11.803 4	12.838 0	13.158 7	13.280 5	变异率:0.05
平均停车次数	7.948 7	33.359 2	4.730 2	13.530 5	交叉率:0.3
每车每路口平均停车次数	0.760 5	2.602 2	0.409 3	1.102 8	世代数:500
每车平均停车时间/steps	29.170 9	48.767 6	12.634 9	23.414 6	$\mu_1 = 1.0$
每车平均运行时间/steps	289.478 6	317.591 5	269.190 5	285.390 2	$\lambda_1 = 0.6$
代价函数均值	0.495 8	1.619 7	0.263 8	0.692 9	$\lambda_2 = 0.4$

注:平时是指仿真时路口未出现大量车辆排队的情况,高峰是指仿真时路口出现大量车辆排队的情况

表 2 数据是仿真系统在相同路网、相同线路结构、相同路口通行规则下仿真得到的结果。从表中可以看出,优化前车辆平均路口停车次数多,平均停车时间长;经过相位优化后车辆平均路口停车次数减少,平均停车时间缩短。尽管优化时产生的车辆具有相同的时空分布,但仿真可按不同方式随机产生车辆,大量仿真表明,优化效果仍然明显,

基本结论不变。

4 结束语

对于城市交通这一复杂系统,通过建立多线路协调优化控制模型,揭示了多条交通线路上车辆与交通信号复杂的时空相位关系,使车辆在此线路上通行时平均停车次数、停车时间有明显减少。可见,通过对各交通路口信号相位差优化,可大大提高现有道路的通行效率。在本模型基础上,采用 TSPE 方法开发了直观的交通仿真系统。相对于国外大型的交通仿真系统模型,本模型具有结构简单、规模小、投资少和实用等特点,有利于用最小的代价成本疏导城市交通,提高城市道路通行能力。由于影响城市交通的因素非常复杂,本文虽然对模型进行了一些简化处理,但在此模型中并没有考虑车辆变道和驾驶员的驾车习惯等问题,模型也未能结合信号配时优化及实时在线检测技术等。这些方面还需要进一步的研究与完善。

参考文献:

[1] 刘智勇.智能交通控制理论及其应用[M].北京:科学出版社,2003.

[2] 全永策.城市交通控制[M].北京:北京人民交通出版社,2004.

[3] 臧志刚,陆锋,李海峰,等.7种微观交通仿真系统的性能评价与比较研究[J].交通与计算机,2007,25(1):66-70.

[4] 朱文兴,贾磊.主干路交通信号控制仿真研究[J].系统仿真学报,2005,17(12):3060-3064.

[5] 朱文兴,贾磊.城市交通主干路相位差优化研究[J].计算机工程与应用,2005,41(20):212-214.

[6] 朱文兴.城市交通系统智能优化控制研究[D].济南:山东大学,2006.

[7] 李晓红.城市干线交通信号协调优化控制及仿真[D].大连:大连理工大学,2007.

[8] 孟维伟,曹从咏.改进时空步长一维交通流的数值模拟[J].交通与计算机,2006,24(3):14-16.

[9] 董超俊,刘智勇,邱祖廉.城市交通控制智能优化配时及仿真[J].系统仿真学报,2005,17(2):472-475.

[10] YANG Li-cai, LIN Jie, WANG De-wei, et al. Dynamic route guidance algorithm based on artificial immune system[J]. Journal of Control Theory and Applications, 2007, 5(4):385-390.

[11] WEN W. A dynamic and automatic traffic light control expert system for solving the road congestion problem[J]. Expert Systems with Applications, 2008, 34(4):2370-2381.

(上接第 2961 页)

[6] GRACE P, LAGASSE B, TRUYEN E, et al. A reflective framework for fine-grained adaptation of aspect-oriented compositions[C]//Proc of the 7th International Symposium on Software Composition. Berlin: Springer-Verlag, 2008: 215-230.

[7] MONDÉJAR R, GARCÍA P, PAIROT C, et al. Building a distributed AOP middleware for large scale systems[C]//Proc of Workshop on NAOMI. New York: ACM Press, 2008:17-22.

[8] MONDÉJAR R, GARCÍA P, PAIROT C, et al. Designing a distributed AOP runtime composition model[C]//Proc of the 24th Annual ACM Symposium on Applied Computing. New York: ACM Press, 2009: 539-540.

[9] WANG Bin, DING Jian, SHENG Jin-fang. Research on model of dual core aspectual middleware & dynamic weaving technology[C]//Proc

of the 9th International Conference for Young Computer Scientists. Washington DC: IEEE Computer Society, 2008: 1035-1040.

[10] WANG Bin, WEN Chun-yan, ZHU Wei, et al. AWright-ADL for aspectual component composition platform[C]//Proc of the 2008 International Symposium on Computer Science and Computational Technology. Washington DC: IEEE Computer Society, 2008: 383-386.

[11] 王斌,谭云桥,桂卫华,等.方面化双核中间件模型[J].计算机工程与科学,2008,30(3):142-144.

[12] 王斌,周亮,谭云桥,等.基于类修改和反射的动态方面编织模型[J].计算机工程与应用,2008,44(7):82-85.

[13] 王斌,周亮,桂卫华.方面编织策略研究进展及图形化方面编织框架[J].计算机工程与应用,2007,43(33):92-97.