

基于发布订阅模式的数据集成中间件系统设计实现^①

Design and Implementation of Publish/Subscribe Schema Based Data Integration Middleware

虞海江 (中国科学院 软件研究所 软件工程技术中心 北京 100080)
 (中国科学技术大学计算机系 安徽合肥 230026)
 马建刚 (中国科学院 软件研究所 软件工程技术中心 北京 100080)
 (中国科学院 研究生院 北京 100049)
 叶丹 李建峰 (中国科学院 软件研究所 软件工程技术中心 北京 100080)

摘要:随着企业信息化的发展,对数据集成系统提出了大规模、动态改变和按需集成数据的需求。针对该应用需求,本文借鉴发布/订阅技术的思想,提出了一种基于发布/订阅的数据集成框架,并设计和实现了基于发布/订阅的数据集成系统。该系统由两部分构成:具有发布订阅功能的数据集成服务器和代理服务器。该方案提供了一种松散耦合的柔性化集成方式,系统应用表明它能适用于大规模的数据集成。

关键词:发布/订阅 数据集成 元数据

1 引言

基于 ETL 的数据集成中间件方案,实现了跨数据源的数据抽取、转换和装载,但这种方案由于采用基于点对点的关联配置的集成方式(如图 1 所示),导致应用时受限于数据集成的规模。每进行一次数据集成,就需要数据的提供者和接收者分别进行手工配置,包括:手工建立发送数据源、发送任务、发送渠道、接收数据源、接收任务、接收渠道和关联映射;当数据提供者的配置发生变化时需要更改相应接收者的配置,进行重新映射。当集成结束时同样需要进行相应配置手动取消。因此,当前基于点对点的关联配置的集成方式有以下两点不足:

(1) 数据提供者和接收者紧耦合,不利于系统扩展。

(2) 对于参与方多、感兴趣的数据也不尽相同,数据需求经常变更的情形,关联配置的维护比较繁琐。

发布/订阅技术与传统的通信范型(消息传递、RPC/RMI 和共享空间)相比,所具有的一系列优点,包

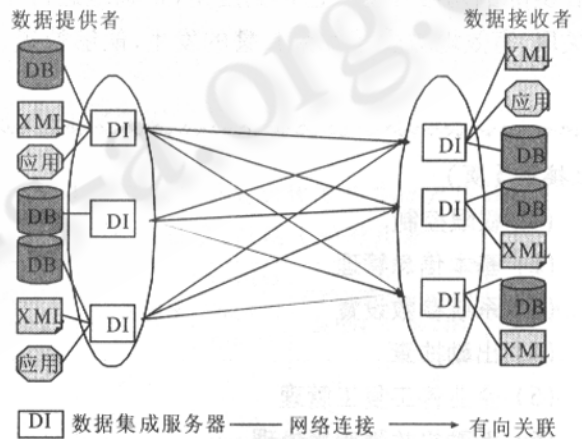


图 1 点对点的关联配置的数据集成方式

括松散耦合、多点通信、匿名性、可扩展性等,使得它成为一种用于大规模数据分发的理想通信方式^[1]。借鉴发布/订阅系统的思想,本文提出了基于发布/订阅方式的数据集成中间件的框架结构,并设计开发了系统

^① 基金项目: National Science Foundation of China under Grant No. 60573126 (国家自然科学基金); Grant No. 2006BAH02A00 (国家科技支撑计划); Grant No. 2007AA04Z148 (国家高科技计划(863))

OnceDI/PS,用以满足当前大规模、动态改变和按照用户需要实现数据集成的需求。

2 基于发布/订阅的数据集成的框架结构

针对当前大规模、动态改变和按照用户需要实现数据集成的需求,我们提出了一种基于发布/订阅的数据集成系统的框架结构(如图 2 所示)。

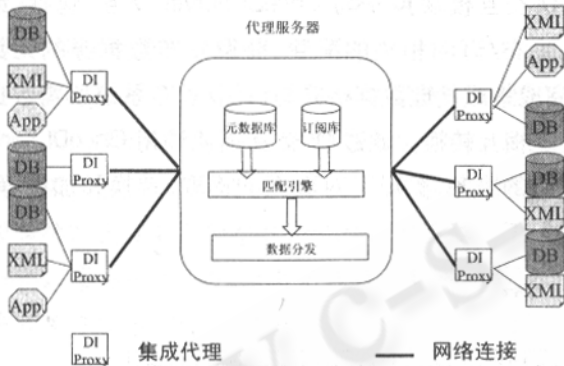


图 2 基于发布/订阅的数据集成的框架

该框架由两部分组成:代理服务器和具有发布/订阅功能的数据集成代理 (DIProxy)。代理服务器负

取消订阅和加载数据等操作。基于发布/订阅的数据集成方式同现有的点对点的数据集成方式相比有以下特点:

(1) 提供了柔性化的集成方式。数据发布者只需指定要发布的数据源及其属性,即可完成数据的发送,无需关心接收者的配置工作;数据接收者只需指定数据过滤条件,以及可选的加载入库方式,即可进行数据的订阅,无需关心数据的发布者配置变化的影响。同现有强关联配置的数据集成方式相比,降低了用户关联配置的维护代价。

(2) 细粒度过滤。数据接收者通过订阅,指定细粒度过滤条件表达对感兴趣的数据的约束,从而获取用户真正感兴趣的数据。

(3) 支持一对多的有选择性数据分发,支持多对一的数据订阅。

当数据发布到代理服务器上后,代理服务器进行订阅匹配,对于符合多个订阅条件的数据,代理服务器自动将其进行分发,从而无需任何配置实现了一对多的数据分发功能;同时对于来自不同发布者符合同一订阅条件的数据,服务器自动将其发往同一订阅者,从而实现了多对一的数据集成。

(4) 提高系统的可用性。用户可以运行时动态的提交元数据发布、订阅及退订请求,无需手动维护配置关联,提高了系统的可用性。

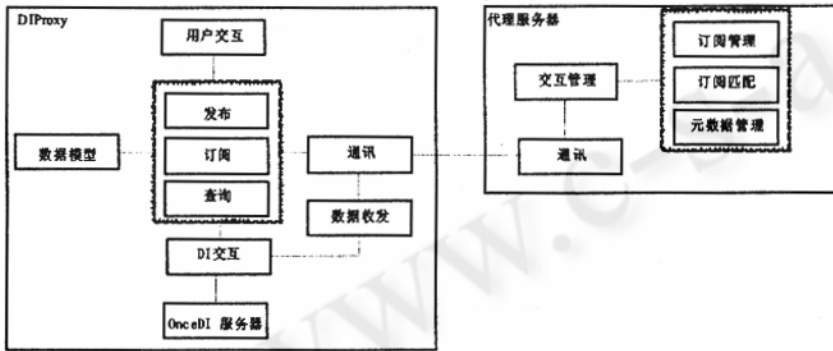
3 基于发布/订阅的数据集成中间件

系统设计

3.1 系统的体系结构

我们构建了基于发布/订阅的数据集成系统 OnceDI/PS (OnceDI Publish/Subscribe)。OnceDI/PS 包括两类实体:代理服务器和具有发布/订阅功能的数据集成代理 (DIProxy)。系统的体系结构如图 3 所示。

集成代理子系统分为三个功能层次:用户交互



于发布/订阅的数据集成系统结构

责存储元数据和订阅、订阅匹配和数据分发。数据的提供者称为发布客户端,数据的使用者称为订阅客户端。DIProxy 具有两种角色:数据发布客户端与订阅客户端。发布客户端连接代理服务器,将从数据源抽取到的元数据、数据信息发布到代理服务器上。订阅客户端连接代理服务器,进行元数据的浏览、注册订阅、

层、功能层和服务层。用户交互层用来处理用户请求,根据请求类型提交给相应的功能模块处理;功能层实现的功能包括:发布请求的处理、订阅请求的处理和查询请求的处理,它们通过与服务层 DI 交互模块、数据模型模块和通信模块的协作来完成;在服务层 DI 交互模块调用 OnceDI 服务器的服务,并与数据模型模块和通信模块协作,为功能层提供数据集成服务。三层的系统结构降低了系统的耦合度,提高了可扩展性。

用户交互模块用于完成系统与用户的交互功能,将用户的操作指令翻译成一系列请求,触发相应的模块执行,从而完成用户所请求的功能,包括:发布请求、订阅请求和查询请求等。

通讯模块为 DIProxy 与代理服务器提供消息通讯服务。提供统一的网络通讯服务。数据模型模块用于建立订阅、事件及元数据,提供统一的数据模型服务。

数据收发模块用于实现数据发布功能和数据接收功能:① 将来自发布数据源的数据转化为系统内部格式的数据,向代理服务器发布;② 接收来自代理服务器的数据,并根据订阅时指定的加载策略把数据加载到相应的数据源。

DI 交互模块用于与 OnceDI Server 交互,包括:维护与发布/订阅相关的配置;获取发布数据源的元数据;完成集成代理数据格式与订阅/发布系统所需数据格式的相互转换。通过 DI 交互模块调用 OnceDI Server 提供的服务,实现了对数据的提取、转换和加载等,

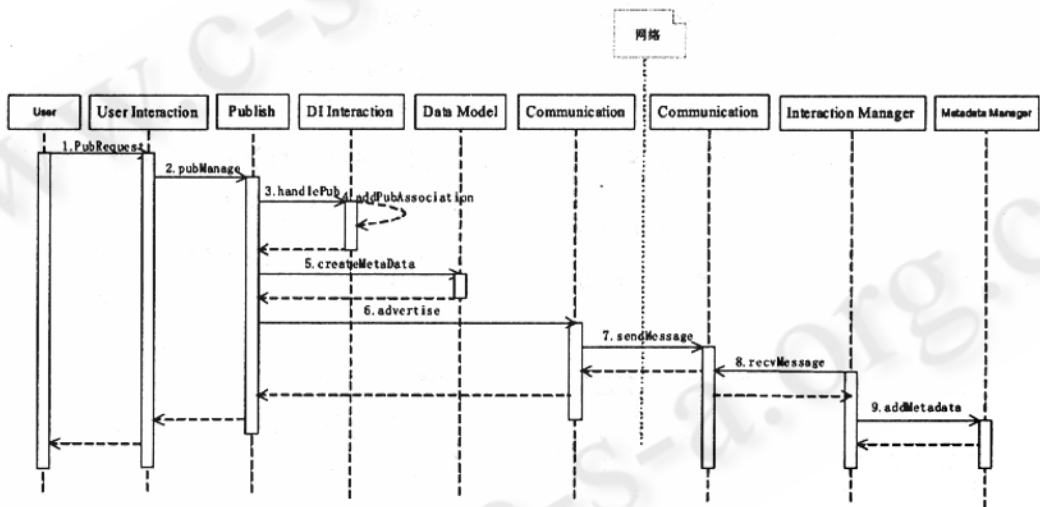


图 4 系统元数据发布顺序图

发布模块、订阅模块和查询模块是 DIProxy 的主要功能模块,用于处理用户请求。发布模块完成发布请求的处理;通过与 DI 交互模块协作,生成与发布相关的 DI 配置,获取发布数据源的元数据信息;通过与数据模型模块的协作,创建元数据;通过与通讯模块的协作,将元数据发布到代理服务器上。订阅模块完成订阅/退订请求的处理;它通过与 DI 交互模块协作,生成/删除与订阅加载策略的相关 DI 配置;通过与数据模型模块的协作,根据订阅请求信息创建订阅;通过与通讯模块的协作,将订阅发送到代理服务器上。查询模块完成订阅和元数据的查询请求的处理。

集成各种异构数据库、文本文件、XML 文件和应用程序等数据资源。

代理服务器存储订阅信息和元数据信息,提供元数据发布与查询、数据订阅与退订的处理、事件的匹配与分发等功能。代理服务器子系统由交互管理模块、订阅管理模块、订阅匹配模块、元数据管理模块、数据模型模块和通讯模块等组成。其中,订阅管理模块负责维护来自集成代理的订阅信息;当收到新的订阅消息时,订阅管理模块负责将其进行预处理;当收到取消订阅消息时,该模块查找相应订阅消息,并将其从维护的订阅信息中删除。元数据管理模块负责维护来自

DIProxy 的元数据信息,并提供元数据的查询功能。订阅匹配模块负责查找与事件(从数据源抽取的数据以事件的方式发布到代理服务器)匹配的订阅,并返回这些订阅的订阅者信息。交互管理模块用于获取通讯模块收到的消息,并根据消息类型分发给订阅管理模块、订阅匹配模块或元数据管理模块执行相应的处理;对于事件消息,通过订阅匹配模块获取订阅者信息后,交互管理模块根据订阅者信息与通讯模块协作完成事件的分发。

3.2 系统的工作流程

集成代理 DIProxy 作为用户与代理服务器进行信息交互的客户端,从角色上看 DIProxy 既可以充当发布客户端,又可以作为订阅客户端。系统的工作流程主要包括对元数据、订阅和事件的处理流程。图 4、图 5 分别以顺序图的形式给出了系统元数据发布、订阅发布和事件发布/接收的流程。

在图 4 中,发布客户端操作如下:

- (1) 用户发出发布请求,并等待发布结果返回;

如:任务、渠道、执行属性等,并保存;获取发布数据源的元数据信息,返回元数据信息;

- (5) Publish 模块将 4) 中得到的结果,提交到 Data Model 模块请求创建元数据,并等待创建结果;

- (6) Publish 模块将 5) 中得到的结果,提交到 Communication 模块进行元数据发布,并等待发布结果;

- (7) Communication 模块,将相应的数据进行打包发送到 BrokerServer 上。

在图 4 中,代理服务器操作如下:

- (8) Interaction Manager 模块获取 Communication 模块收到的消息,并等待获取结果;

- (9) Interaction Manager 模块,根据消息的类型将元数据消息提交 Metadata Manager 模块处理,并等待处理结果。

在图 5 中,订阅客户端操作如下:

- (1) 用户发出订阅请求,并等待订阅结果返回;
- (2) User Interaction 模块,验证请求的合法性;负

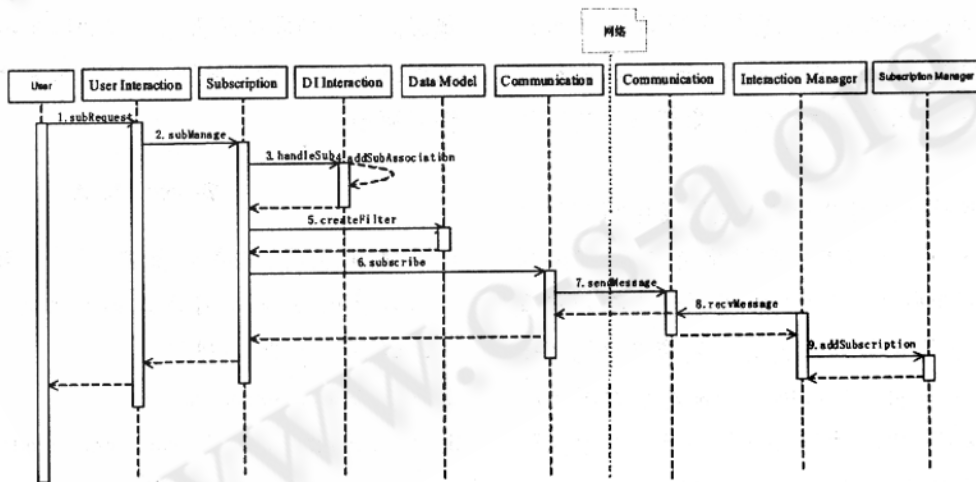


图 5 系统订阅发布处理顺序图

- (2) User Interaction 模块验证请求的合法性;负责向 Publish 模块提交发布请求,并等待发布结果;

- (3) Publish 模块解析发布请求,将发布数据源和发布属性提交到 DI Interaction 模块,并等待处理结果;

- (4) DI Interaction 模块完成对发布数据源的合法性检查,生成发布 ID,创建发布请求的相关配置和关联

负责向 Subscription 模块提交订阅请求,并等待订阅结果;

- (3) Subscription 模块解析订阅请求,将订阅条件和所选的订阅加载策略,提交 DI Interaction 模块,并等待处理结果;

- (4) DI Interaction 模块,检查订阅条件的合法性;

生成订阅 ID,根据订阅条件和所选的订阅加载策略,生成相应的订阅关联并保存;返回订阅 ID;

(5) Subscription 模块检查 4)中得到的结果,将订阅条件和订阅 ID 提交给 Data Model 模块请求创建订阅,并等待创建结果;

(6) Subscription 模块将 5)中得到的结果,提交给 Communication 模块进行数据订阅,并等待订阅结果;

(7) Communication 模块,将相应的数据进行打包发送到 BrokerServer 上。

在图 5 中,代理服务器操作如下:

(8) Interaction Manager 模块获取 Communication 模块收到的消息,并等待获取结果;

(9) Interaction Manager 模块,根据消息的类型将订阅消息提交 Subscription Manager 模块处理,并等待处理结果。

4 系统实现与应用

我们在中科院软件所自主开发的数据集成中间件 OnceDI^[2]的基础上,开发了支持发布/订阅方式的数据集成中间件系统 OnceDI/PS。它提供了跨平台、跨数据源的异构数据集成,实现了数据提取、转换、传输和加载的全过程操作,提供了对发布/订阅集成方式的有效支持。

该系统在普通话语音教学和考试系统中得到了应用。由多个教学中心组成的发布者集合,通过定期将不同的语音数据信息通过 OnceDI/PS 系统发布到代理服务器;由不同教学单位组成的订阅者集合通过 OnceDI/PS 订阅感兴趣的语音数据。于是对于教学中心和教学单位只需将数据的发布请求和订阅请求发送到 OnceDI/PS 系统,既可以自动完成语音数据的发送和接收。实际应用表明 OnceDI/PS 系统可以有效的支持大规模数据集成的应用。

5 相关工作比较

目前,主要的数据集成方案包括联邦数据库、数据仓库和数据集成中间件等。联邦数据库和数据仓库方案存在的问题是数据集成局限于已有主题,全局模式

难以构建^[3]。

而基于 ETL 的数据集成中间件方案,实现了跨操作系统、跨数据源的数据抽取、转换和装载,使得数据集成独立于主题,在企业应用中具有更多的灵活性和适应性^[4]。OnceDI/PS 是一种支持发布/订阅方式的 ETL 数据集成中间件系统,提供了松散耦合的数据集成方式,减少了用户关联配置的负担,实现了多对一的数据集成。

Hermes 系统^[5]把发布/订阅技术应用到主动数据库上,但是局限于同构的数据库。我们的方案支持异构数据库的集成。

6 结束语

本文借鉴发布/订阅技术的思想,提出了一种基于发布/订阅的数据集成框架,设计并实现了系统 OnceDI/PS。该系统由两部分构成:具有发布订阅功能的数据集成代理和代理服务器。它提供了松散耦合的数据集成方式,减少了用户关联配置的负担,实现了多对一的数据集成。系统应用表明它能适用于大规模的数据集成。下一步准备研究布式代理服务器网络环境下的事件、订阅和元数据的路由策略,来避免集中式结构的性能瓶颈。

参考文献

- 1 马建刚、黄涛、汪锦岭、徐昱、叶丹,面向大规模分布式计算的发布订阅系统的核心技术,软件学报,2006,17(1),134-147.
- 2 <http://www.once.org.cn/>
- 3 A. P. Sheth, J. A. Larson. Federated database system for managing distributed, heterogeneous, and autonomous database. ACM Computing Surveys, 1990, 22(3): 183-236.
- 4 钟华、冯文澜、谭红星等,面向数据集成的 ETL 系统设计与实现,计算机科学,2004,31(9):87-90.
- 5 P. R. Pietzuch, "Hermes: A Scalable Event-Based Middleware", PhD thesis. University of Cambridge, 2004.