

基于网格技术的物联网 Savant 中间件的实现技术*

潘 林, 赵会群, 孙 晶

(北方工业大学 信息工程学院, 北京 100041)

摘要: 简述物联网 Savant 中间件的工作原理, 分析了目前 Savant 系统中存在的缺陷, 并提出了相应的解决办法。讨论了网格技术的特点及与构造 Savant 系统紧密相关的技术细节和系统构建思想, 就基于网格技术构造可扩展的 Savant 系统进行研究。提出了一种基于网格技术的 Savant 系统体系结构, 并分析了这个体系结构的特点及优势。详细介绍了 Savant 系统的内部模块功能和交互方式, 并介绍了一个能够满足这种设计需求的 Savant 系统原型的功能特点及实现方法。

关键词: 物联网; Savant 中间件; 网格技术

中图分类号: TP311.11

文献标志码: A

文章编号: 1001-3695(2007)06-0292-03

Implementation Technology of Savant Middleware in EPC Network Based on Grid

PAN Lin, ZHAO Hui-qun, SUN Jing

(College of Information Engineering, North China University of Technology, Beijing 100041, China)

Abstract: Expatiated the working principle of the Savant middleware and analyzed the lacunas in the present Savant system and gave a solution about it. Discussed the features of the grid technology, the detail about this technology and the idea of the Savant system construction. After discussing the topic about constructing the extensible Savant system, a new architecture of Savant system base on the grid was presented. It analyzed the features of the system, introduced the inner models of the Savant system and the mode of communicate with each other. Base on it, a Savant prototype system was designed and implemented.

Key words: EPC network; Savant middleware; grid

0 引言

基于互联网和射频技术的 EPC (Electronic Product Code) 系统, 即实物物联网 (物联网), 是在计算机互联网的基础上利用射频识别 (Radio Frequency Identification, RFID)、天线通信等技术, 构造了一个实现全球物品信息实时共享的平台。它将成为继条码技术之后, 再次变革商品零售结算、物流配送及产品跟踪管理模式的一项新技术。通过 EPC 系统不仅能够对货品进行实时跟踪, 还能通过优化整个供应链给用户提供支持, 从而推动自动识别技术的快速发展并能够大幅度提高全球消费者的生活质量。

Savant 是一个物联网系统的中间件, 用来处理从一个或多个解读器发出的标签流或传感器数据 (事件数据)。Savant 是连接标签解读器和企业应用程序的纽带; 代表应用程序提供一系列计算功能。在将数据送往企业应用程序之前, 它要对标签数据进行过滤、总计和计数, 压缩数据容量。为了减少网络流量, Savant 也许只向上层转发它感兴趣的某些事件或事件摘要^[1]。

在 IBM、Sun、沃尔玛等全球 83 家跨国公司的支持下, 2003 年完成了 EPC 系统技术的规模场地使用测试。2005 年, EPC-global 正式颁布了 EPC Network 蓝本^[2]。其中将 Auto-ID Sa-

vant Specification 1.0^[3] 作为 Savant 中间件构造的参考实现规范。根据相关规范, 包括 IBM、Sun 等国际著名 IT 公司, 均在自己的技术基础上, 提出了相应的 EPC Network 解决方案, 并且在包括航空、物流、生产制造等多个领域投入实际应用, 取得了一定的商业效果。物联网技术在我国起步较晚, 但是发展很快。以广州、上海为首的大中城市, 均开始了 EPC Network 的构建和推广工作, 并且有了一些成功的商业案例。

但是, 作为 EPC Network 的关键组件, Savant 模块的构建和实现始终没有一个完善的解决方案。主要体现在以下几点:

(1) 系统的封闭性。各大厂商对 Savant 甚至 EPC Network 所提出的解决方案, 均与自己已有的技术紧密耦合。这使得构建出的系统与外部世界缺乏交互能力。在 Web Service 技术和网格平台大量应用的状况下, 这种构建封闭式系统的方式制约了系统的扩展。

(2) 系统实现的不确定性。Savant 规范在一些部分, 如 Savant 之间的通信方式等方面还没有作出完善的规定^[3], 使得对规范的理解和实现出现困难。

解决上述不足的一个有效办法是实现基于网格平台的 Savant 系统, 屏蔽系统实现的内部技术细节; 充分利用 Web Service 技术, 采用统一的通信方式; 实现系统良好的可扩展性。

在多数 Savant 系统中, 包括了边界节点 (Edge Savant) 和内部节点 (Internal Savant); 节点多为异地分布。系统采用典型

的分布式计算方式。网格技术作为 SOA (Service-Oriented Architecture) 思想的典型体现,为构建分布式系统提供了底层技术支持。网格的主要任务是在动态变化的网络环境中共享资源和协同解决问题。网格对现有互联网进行了非常好的管理,它将分散在不同地理位置的资源虚拟成一个整体的信息系统,实现计算资源、存储资源、数据资源、信息资源、软件资源、通信资源、知识资源和专家资源等的全面共享。GT4 (Globus Toolkit 4) 则为分布式系统的构建提供了成熟的平台。

在具体应用上,基于 GT4 研发的许多系统均取得了良好的效果^[4]。作为分布式计算思想的具体应用,Savant 系统的实现可充分利用 GT4 提供的许多服务资源,有效管理系统内部资源,协调与外部系统之间的交互。基于 SOA 思想对系统进行设计和实现,使得系统具有良好的可扩展性;通过对 Web Service 技术的应用,可实现 Savant 内部各个节点间以及与外部各组件间通信方式的统一;对 WSRF (Web Service Resource Frame) 规范的实现,使得基于 GT4 平台可以提供有状态的服务。利用这一特性,内部节点可以保存当前的相关状态,供外部的服务消费模块调用和修改,方便了外部系统对 Savant 模块的控制。

1 基于网格技术的 Savant 系统方案

在现有的相关标准基本确定的情况下,实现一种可扩展的 Savant 系统并实现相关标准设定的业务逻辑,有效的途径是改进 Savant 系统体系结构;同时以网格作为基础平台,采用 SOA 的设计思想。

1.1 可扩展的新型 Savant 系统体系结构

Savant 系统由边界节点 (ES) 和内部节点 (IS) 构成。其中内部节点间多为异地分布。这一特点决定了分布式的系统架构应该是一种必然的选择。基于网格技术的体系结构则能够最大限度地发挥分布式计算的优势。为此,笔者提出了这种新型的 Savant 体系结构 (图 1)。

从层次上看,Savant 系统的体系结构是一种树型结构。边界节点从属于作为其父节点的内部节点,而内部节点之间也存在父子关系。边界节点与 RFID 标签的解读器相连。解读器不停地从标签中采集 EPC 数据,并向 Savant 传输。每次识读,Savant 都要保存相关信息,内部节点从其下属边界节点中采集 EPC 数据,还负责合计 EPC 数据;并且与外部的企业信息系统、信息服务系统以及 ONS (Object Name Service) 服务系统进行通信。

从交互性上看,由于边界节点与其父节点会在同一个地点,这两者间并不建立 Web Service 通信,保证了信息处理速度;内部节点与其父节点建立通信联系,通过网格环境提供的各种服务,最大限度地保证信息安全和计算资源的高效利用;内部节点与外部环境之间通过 Web Service 相连,保证了当需要进行业务逻辑调整时系统架构有良好的可扩展性。

1.2 网格技术的应用

作为当前最流行的系统架构思想,SOA 思想提倡利用显式的与实现无关的接口来定义服务。利用强调位置透明性和

可操作性的通信协议,封装可重用业务功能服务的定义^[5],为系统的可扩展性提供了良好的支持。网格技术作为 SOA 思想的典型实现,为搭建面向服务的分布式系统提供了良好的技术支持。网格技术在理论上进一步成熟的同时,GT4 对新的网格技术规范提供的完善的实现,为系统的搭建提供了技术平台。

本文的 Savant 系统需要提供有状态的 Web Service,用于协调系统内部各类资源并保存相应的状态^[6]。对于 Savant 内部节点间的交互操作,可以充分利用 GT4 对 WSRF 的实现,维护 Web Service 交互时服务的状态,从而确保对各个节点状态的控制和调整;同时,通过 Web Service 实现基于 SOAP 的消息传输机制,统一了内部节点间的通信格式。

2 基于网格技术的 Savant 系统的设计与实现方法

本文的 Savant 系统通过利用 GT4 对 WSRF 规范的实现,达到了在 Web Service 环境下保持系统相应状态的目的,为外部系统对 Savant 模块进行交互提供了便利。下面对本文的 Savant 系统的功能结构进行阐述。

2.1 功能结构

边界节点中,RFID 阅读器用于接收 RFID 标签的射频信息。信息由数据处理模块初步处理后,通过数据传输接口向相应的内部节点传送。即时内存数据结构 (RIED) 模块能够在保证存取速度的情况下,作为上传和下载数据的缓存。核心模块提供边界节点的相关信息。控制接口通过这些信息,按照上层内部节点的命令,对边界节点的业务动作进行控制 (图 2)。

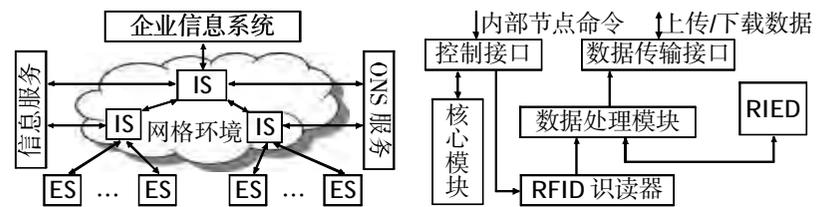


图 1 新型的 Savant 体系结构

图 2 Savant 系统边界节点

内部节点由事件管理系统 (EMS)、任务管理系统 (TMS)、接口模块,以及 RIED 和核心模块组成 (图 3)。

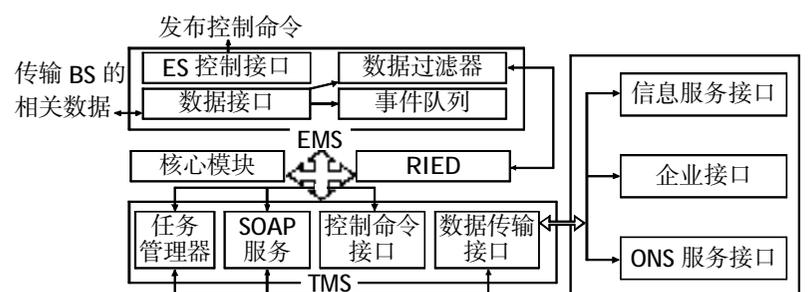


图 3 Savant 系统内部节点

事件管理系统 (EMS) 由边界节点 (ES) 控制接口、数据过滤器、数据接口以及事件队列组成。数据接口和数据过滤器用于接收和处理从边界节点得到的信息。事件队列作为事件流的缓存,是一个异步排队系统,但采用同步机制处理多种解读器事件记录。为保证信息传输的速度,EMS 与边界节点之间并不采用 Web Service 的方式进行联系,而是通过互通 IP 地址直接相连。

任务管理系统 (TMS) 由数据传输接口、控制命令接口、任务管理器以及 SOAP 服务模块组成;负责管理由上级 Savant 或

企业应用程序发送到本级 Savant 的任务。一般情况下,一个任务可以等价于多任务系统中的进程。TMS 使分布式 Savant 的维护变得简单。企业可以仅仅通过在一组类服务器保存最新任务和在 Savant 中恰当地安排任务进度来维护 Savant。任务管理器负责运行和维护 Savant 中的任务。这些任务代表用户在 Savant 端运行。SOAP 服务器负责表明任务管理器的功能和接口,它使任务管理器成为可以被所有系统统一访问的 SOAP 服务。通过提交包含任务管理器表述详细办法的简单配置描述符文档,可以获得接口配置。随着任务管理器的成熟及其 API 的扩展,可通过增加新表述办法来升级描述符。

接口模块由面向企业的接口、信息服务接口以及 ONS 服务接口构成。它是 Savant 系统与外界交互的通道。基于网格技术,接口模块向外部环境提供有状态的 Web Service,屏蔽了系统内部实现的差异,统一了系统与外部环境之间的通信方式。由于采用 SOA 的设计思想,系统变得灵活而易于扩展。当外部环境的业务逻辑改变时,只要 Web Service 接口不变,Savant 就无须作调整。将 GT4 作为基础平台,能够充分利用 WSRF 对 Savant 系统的资源进行协调。

2.2 系统实现方法

根据上文的分析和实际需求,实现这样一个分布式的 Savant 系统需要考虑并解决以下问题:

(1) 系统的响应速度。Savant 系统中,尤其是边界节点,一般部署在超市、仓库等地点,对信息处理响应的速度要求比较高。为此,系统采用了内存数据库,提高数据的存取速度,通过这些边界节点的直接父节点循环读取它们内存数据库中的相应数据,将这些 RFID 数据读入内部节点,作进一步处理。同时,系统采用缓存策略,将热门商品信息提前缓存在本地,最大限度地提高了系统的响应速度。在内部节点中,通过设计良好的数据结构,实现事件队列和任务队列,实现事件流和任务流的缓存。

(2) 服务设计的粒度。系统的内部节点暴露的服务接口的基于网格技术,作为 SOA 思想的具体应用,需要考虑服务粒度的设计。系统采用将服务分类的策略,对内部节点之间使用的服务,采用较细粒度的设计方式,以保证内部节点间控制功

能的细致性。本文设计的系统在内部节点之间的交互不会涉及大量数据的交互,所以采用 SOAP 进行消息传递;同时,对于向外部环境暴露的服务采用粗粒度的设计方式,保证业务逻辑的清晰性,以及扩展业务逻辑时的便捷性。

(3) 客户端升级问题。Savant 系统的边界节点通常需要提供显示界面供用户操作。某些内部节点也需要提供用户界面以便于管理。随着业务逻辑的改变或系统的升级,客户端的升级问题必须考虑。系统采用 Java Swing 技术向用户提供客户端界面,并采用 Java WebStar 技术解决客户端在线升级问题,使得系统的更新更加便捷。

3 结束语

本文简述了物联网的工作原理,提出了一个物联网 Savant 中间件的系统模型;以 GT4 作为基础平台,采用 SOA 的设计思想,在实现了 Savant 规范的同时,达到了系统良好的可扩展性。同时,通过良好的设计,系统达到了良好的响应速度,解决了客户端实时升级的问题。本文给出了系统实现的方法,以及在设计实现过程中需要注意和解决的问题,给出了将网格技术和 Savant 规范的实现相结合的方法。

参考文献:

[1] 中国物品编码中心. Savant 技术说明书 [EB/OL] . <http://www.rfidworld.com.cn/bbs/>.

[2] EPC Global. The EPCglobal architecture framework [EB/OL] . <http://www.epcglobalinc.org>.

[3] Auto-ID Center. Auto-ID savant specification 1.0 [EB/OL] . <http://www.epcglobalinc.org/>.

[4] FOSTER I, KESSELMAN C. The grid 2—blueprint for a new computing infrastructure [M] . 北京:电子工业出版社, 2004: 37.

[5] KOROTKIY M, TOP J. Onto-SOA: from ontology-enabled SOA to service-enabled ontologies, telecommunications: AICT-ICIW '06 [C] . [S. l.] : IEEE, 2006.

[6] UPPULURI P, JABISETTI N, JOSHI U, et al. Service oriented framework for distributed resource management: proc. of International Conference on Services Computing [C] . [S. l.] : IEEE, 2005.

(上接第 291 页) 思路是通过审批流程模板模块和公文审批模块来实现建立的公文审批系统模型。通过接口与 PDM 系统的其他功能相连,基本能够满足企业对公文安全和快速审批的要求。



图 7 公文监控工具

参考文献:

[1] OLLINGSWORTH D. The workflow reference model [EB/OL] . <http://www.wfmc.org/standards/docs/tc003v11.pdf>.

[2] AALST V der, HOFSTED E T, BARROS K. Workflow patterns [J] . Distributed and Parallel Databases, 2003, 14 (1) : 5-51.

[3] CHOI I, PARK C, LEE C. A transactional workflow model for engineering/manufacturing processes [J] . International Journal of Computer Integrated Manufacturing, 2002, 15 (2) : 178-192.

[4] 王少锋,王克宏. 工作流管理系统中同步关系的实现方法 [J] . 计算机集成制造系统, 2001, 7 (3) : 37-41.

[5] 陈石灵,张友良. 工作流管理系统中规则处理问题研究 [J] . 计算机工程与应用, 2001, 37 (16) : 14-16.

[6] 范文慧,李涛,熊光楞. 产品数据管理 (PDM) 的原理与实施 [M] . 北京:机械工业出版社, 2004: 24-25.