

面向 SOA 的企业服务总线研究与应用^①

曹中洪 廉东本 (中国科学院 沈阳计算技术研究所 辽宁 沈阳 110171)

摘要: 针对企业日益增长的信息共享和业务的需要, 部署企业服务总线 (ESB) 集成企业的各类异构应用是必然发展趋势。现有的 ESB 大多数立足于构建消息中间件, 没有全面考虑 ESB 的功能需求, 从而极大的限制了 ESB 的高灵活性和高可扩展性, 不能满足企业的高度并发的访问和快速实时响应的需求。提出了基于插件技术的 ESB 体系结构模型, 并解决了高度并发请求的实时捕获和响应问题; 原型系统在实际项目中的应用证明了 ESB 的高灵活性和高可扩展性。

关键词: 面向服务架构; 企业应用集成; 企业服务总线

Research and Application of Enterprise Service Bus

CAO Zhong-Hong, LIAN Dong-Ben

(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110171, China)

Abstract: For the quickly growing standards of information sharing and business integration in enterprise, SOA is often used to integrate enterprise applications because enterprise service bus can shield the heterogeneous enterprise applications. Current products of Enterprise Service Bus (ESB) are mainly based on Message Middleware technology, which do not satisfy all the requirements of ESB. These products restrict the availability and scalability of ESB, and cannot meet the enterprise's highly concurrent request and real-time response requirements. In this paper, a new architecture is first brought forward for ESB based on Plug-In technology, and then it analyzes some key technologies including capture requests, dispatch requests, real-time response etc. At last, ESB has proven to be feasible and effective by the prototype system's application in practical projects.

Keywords: service oriented architecture (SOA); enterprise application integration (EAI); enterprise service bus (ESB)

1 引言

随着经济全球化以及信息技术的发展, 企业的运营模式随之发生巨大的变化: 突破了地理限制, 跨地区、跨国经营; 不断的兼并与收购, 形成了众多的产品线和业务; 与合作伙伴进行信息共享、协作。这就要求企业构建灵活、高效的应用系统: 消除地理位置的限制; 能够简单便捷的集成新的业务系统; 与合作伙伴之间共享信息资源, 保持紧密的协作关系。只有这样, 才能在瞬息万变的市场上, 保证企业的有效

运营和经济效益。这对企业应用集成提出了新的挑战和需求: 能够越过地理位置的障碍, 统一全面的管理企业运营的各个方面; 在充分利用现有系统的基础上, 灵活有效的整合新业务和新流程; 保证企业的各类异构系统能够互联互通、协同工作; 能够与第三方应用系统进行信息资源共享与协作。

要有效应对企业应用集成的挑战和需求, 就必须实施 SOA(面向服务架构, Service Oriented Architecture)。IBM 公司给 SOA 的定义是: “SOA

^① 收稿时间:2010-01-15;收到修改稿时间:2010-03-18

是一个组件模型,它将应用程序的不同功能单元(称为服务)通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的,它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种这样的系统中的服务可以以一种统一和通用的方式进行交互”。^[1]SOA 只是一种软件系统架构,在实际应用中需要借助面向 SOA 的企业服务总线 (Enterprise Service Bus)。目前,产业界已经推出了诸多面向 SOA 的企业服务总线,如 IBM 公司的 WebSphere Service Bus^[2], BEA 公司 AquaLogic Service Bus^[3], IONA XMLBus^[4]等。这些产品以消息中间件为基础,提供 Web 服务的运行管理环境,不能满足企业服务总线的全部功能需求,灵活性和扩展性受到很大限制。

充分利用 SOA 的高度灵活性、可扩展性以及低耦合性,综合分析服务总线的全部功能需求,提出以插件技术为基础的服务总线体系结构,从体系结构层次上解决应用组件之间的耦合性问题,提高软件系统的复用性、降低系统的复杂性和构建成本;深入研究并解决企业应用系统的高并发访问和实时响应问题,满足企业的苛刻要求,为企业的应用集成和业务整合提供支持 and 保证;最后,原型系统证明了基于插件技术面向 SOA 的企业服务总线具有高灵活性、高可扩展性。

2 体系结构

2.1 ESB 的应用集成

ESB 是消息中间件的进一步发展,采用“总线”这样一种模式来管理应用系统之间的集成拓扑结构,通过统一的标准接口,提供服务注册、服务查找、服务调用等核心功能,从而完成服务之间的交互^[5]。

ESB 的基本特征和能力包括:描述服务的元数据和服务管理;在服务提供者和服务请求者之间提供数据转换、传递的功能,支持多种基本模式,如同步与异步、发布于订阅等;发现服务,路由,匹配和选择服务的能力。

通过 ESB 对外提供标准的连接方法,各种应用可以将其一个或者多个模块封装为服务发布出去。ESB 能够统筹管理所有应用系统发布的服务,进行服务识别、验证以及分类。当某一服务请求到来时,ESB 可以进行服务查找和分配,而 ESB 的内核可以将服务请

求转发到服务提供者,在服务提供者完成任务后,ESB 再将服务提供者返回的结果消息转发到服务请求者,从而实现了服务之间的交互。

ESB 的应用集成环境如图 1 所示:

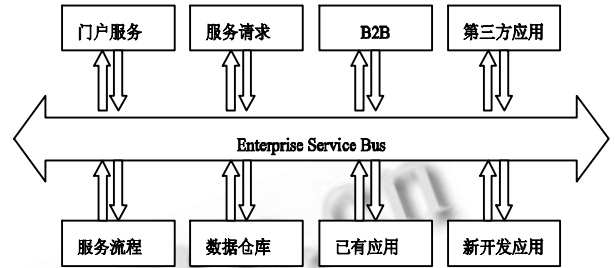


图 1 ESB 应用集成环境

2.2 面向服务架构

企业服务总线是面向服务架构 (SOA) 的一种重要的实现方式。SOA 中有 3 个最重要的角色,分别是:服务请求者、服务代理、服务提供者,3 者之间的关系如图 2 所示。

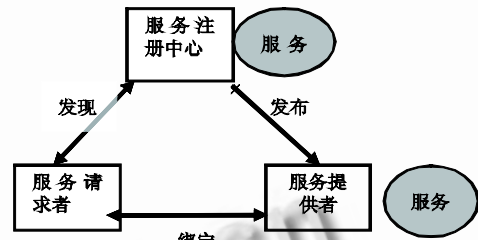


图 2 面向服务架构

SOA 包括如下要素:

- ①一个体系架构,用开放的标准将软件资产以服务的形式展现
- ②提供标准的方法来表示软件资产及其交互
- ③独立的软件资产作为构造单元,被重复使用来开发其他应用
- ④将关注点从细节实现转移到应用组装
- ⑤被内部用于基于既有的组件创建新的应用
- ⑥被外部用于与企业外部应用(B2B)整合
- ⑦开发(现在)和整合(未来)的统一

SOA 并不是一种现成的技术,而是一种架构和组织 IT 基础结构及业务功能的方法。^[6]SOA 是一种在计算环境中设计、开发、部署和管理离散逻辑单元(服务)的模型^[7]。

3 关键技术

3.1 服务请求的捕获

服务请求监听器负责服务请求的捕获。要适应企业服务总线服务请求的高度并发性，服务请求监听器的设计尤为重要。传统的单线程、单端口模式只能以串行模式工作，显然无法满足需求。因此，必须采用多线程与多端口相结合的技术来解决问题。如图3所示，ESB内核可以启动多个服务请求监听器线程，分别对多个端口并行监听，从而有效均衡各个端口的负载提高系统的响应速度。

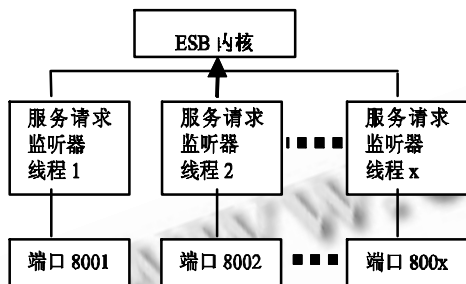


图3 服务请求监听

具体思想如下：

①服务总线系统启动时根据配置信息创建多个服务请求监听器线程，将它们放在线程池中，每个线程对应一个指定的端口。

②服务请求者生成服务请求后，随机选择一个端口发送请求。

③所有监听器线程都处于活动状态，一旦从对应端口接收到服务请求，立即处理。

设有 N 个监听器线程，端口号从 8001 到 800N，单个服务请求的处理时间为 T (均值)，则系统最多可以同时处理 N 个服务请求，系统处理服务请求的平均时间为 T/N ，平均时间随 N 的增大而减小。因此，系统具有很高的并发性，能够满足实时性需求。

3.2 服务请求的派发

服务请求的派发是企业服务总线系统的另一个关键技术问题，如何有效管理和快速派发服务请求直接反应了系统的响应速度。服务请求派发的一般方法是建立一个先进先出(FIFO)的请求消息队列，服务请求消息按照先后顺序派发。这种方法在应对小规模应用时是简单实用的，但是在应对企业级大规模应用时将会出现很多问题：单一的消息队列只能以串行的方式调度，调度效率非常低；消息没有优先级的区分，

难以满足实际需要。因此，必须设计一种多优先级并发的调度模型，如图4所示。

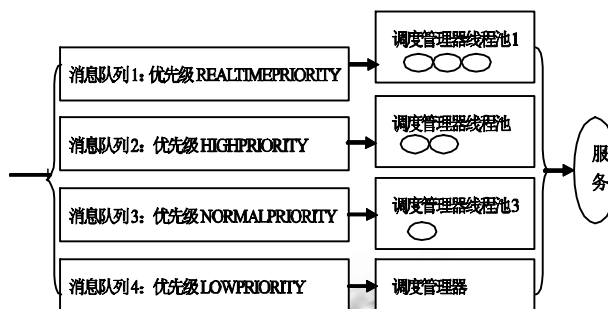


图4 服务请求调度示意图

如图所示，优先级队列共有 4 个，分别是：REALTIMEPRIORITY 实时优先级；HIGHPRIORITY 高优先级；NORMALPRIORITY 普通优先级；LOWPRIORITY 低优先级。每一个优先级对应一个消息队列和一个调度管理器线程池。优先级的高低是通过调度管理器线程池中的线程数量的多少来反映的。线程数量越多，消息队列中的消息处理速度也就越快。按优先级从高到低，线程池中所包含的线程数量之比为：10：6：3：1。多优先级消息队列调度算法代码如下：

```

while(true)
{
    for(k=1; k>0; k--){
        //消息队列 Qk 中有等待调度的请求
        if(MessageQueue[k].size() < 0) continue;

        //对应的线程池中有空闲线程可用
        while(ThreadPool[k].hasVacantThread()) {
            //从消息队列中取出排在最前面的请求
            mes = MessageQueue[k].removeFromHead();

            //从线程池中取出一个空闲线程
            trd = ThreadPool[k].getVacantThread();

            //将取出的请求作为参数，启动该线程执行服务调用，等待调用完成
            while(rst = dispatchService(trd, mes));

            //返回结果通过请求监听器，发送给服务请求者
  
```

```

requestListener.setResult(rst);

//该线程恢复为空闲线程
ThreadPool[k].insertThread(trd);
    }
//当前线程池无空闲线程可用
//转到下一个优先级的队列进行调度，流程同上
    }
}
    
```

4 ESB的实现与测试

4.1 ESB 的实现

ESB 作为企业级的应用系统，必须具有高度灵活性和可扩展性的体系结构，良好的运行性能和容错性，满足企业级应用的高度并发的访问请求，并且具有最快的响应速度。

以插件技术为基础，设计如图 5 所示的 ESB 原型系统。服务注册管理器：提供服务注册请求接口；服务请求监听器：监听服务请求消息；服务调用管理器：调用具体服务；路由器：查找并派发服务请求；ESB 管理工具箱：管理工具的集合，包括用户管理、ESB 运行状态管理、ESB 运行监控、以及 ESB 插件的部署和安装等。数据库包含 3 个部分：服务库存储所有的服务信息如服务名、参数、地址；日志库存储 ESB 运行时记录的所有日志信息；用户库存储 ESB 的用户信息。

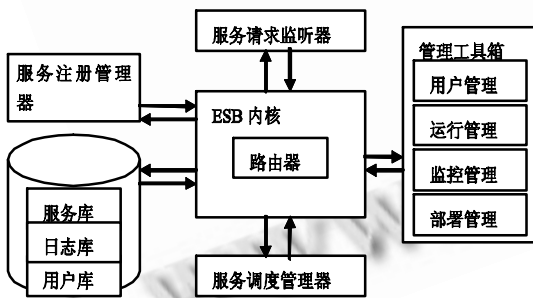


图 5 ESB 体系结构

服务总线的典型运行流程是：服务请求监听器接收到服务请求并转交到路由器，路由器将服务请求派发到对应的服务提供者，然后交由服务调用管理器执行服务调用，调用结束后，结果以相反过程转到服务请求监听器，最终到达服务请求者处。

4.2 ESB 的应用

现有两个应用系统：一、沈阳市安全生产管理 GIS

平台(系统 1)；二、沈阳市安全管理应急平台(系统 2)。系统 1 以 GIS 为基础，实现综合管理煤矿、发电、石油、化工等涉危企业，监控企业的生产状况，并且能够通过 GIS 显示。系统 2 以 MIS(信息管理系统)为基础，对沈阳市的所有危险源进行管理，能够模拟事故并使用人工智能方法提供最有利的应急救援和处置方案。两个系统在实施危险源的管理任务上有多个功能模块需要交互与调用：

- 1) 系统 2 通过系统 1 获得企业的地理位置信息。
- 2) 系统 1 通过系统 2 获得企业的详细信息。
- 3) 系统 2 的应急救援方案通过系统 1 来真实模拟和展现。
- 4) 系统 2 根据系统 1 提供的地理信息，结合人工智能求解最优救援路径和方案。

利用企业服务总线连接系统 1 和系统 2，可以在不对原有系统作较大改动的前提下，实现系统间的交互调用，充分降低了系统之间的耦合性，提供了一个非常灵活的应用集成整合环境，如图 6 所示：

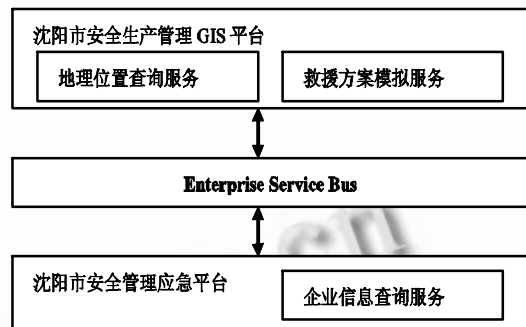


图 6 ESB 的实际应用

4.3 ESB 的测试

为了正确测试 ESB 系统的运行性能，采用多种测试方案，并进行对比分析。以地理位置查询服务为例，测试环境为 Core T2300 1.6GHz/ 内存 2G/Windows 2003/JDK1.5/ ESB1.0，测试结果分别如下：

- 1) 并发发送服务请求，ESB 系统的接收全部请求的所消耗的时间，测试结果如图 7 所示：

图中直线为理想情况下，请求速率与消耗时间之比，红线为实际测试结果。由图可知，实际运行结果在理想直线的上方，有轻微的波动变化。原因是随机算法导致各个端口接收到的请求数量不同，实际消耗的时间是接收到请求最多的端口所消耗的时间。从总

体来看,消耗的时间与理想时间相差不大,服务请求监听的性能达到了目标要求。

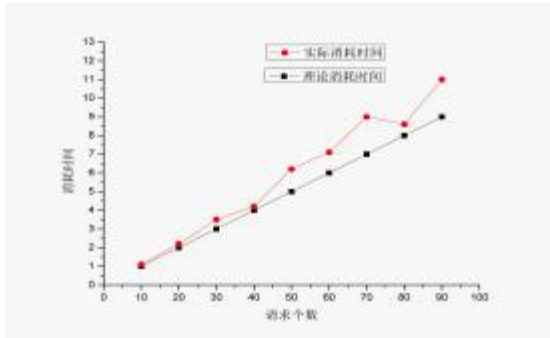


图7 测试结果1

2) 分别取 REALTIMEPRIORITY、HIGH PRIORITY、NORMALPRIORITY、LOWPRIORITY 四种优先级,发送速率分别为 20、40、60、80,测试结果如图8所示:

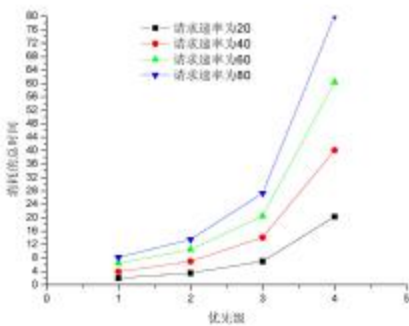


图8 测试结果2

优先级从高到底依次为 1 到 4。从图中可以看出,高优先级的请求得到处理所消耗的时间较低优先级请求所消耗的时间有非常大的差别;高优先级的请求能够在最短时间内得到处理,低优先级的请求处理速度

相对较慢。测试结果证明了采用多优先级队列调度的方法能够满足需求。

5 结语

企业服务总线是集成企业各类异构应用的基础技术和核心构件。现有的企业服务总线普遍是以消息中间件为基础,难以满足企业服务总线的全部功能需求,设计并实现了以插件技术构建的企业服务总线,深入研究并解决了并发访问的实时响应问题。最后通过原型系统证明了插件技术构建的企业服务总线具有高灵活性、高可扩展性。

参考文献

- 1 Patterns: Service-oriented architecture and Web service. [2009-11-02].<http://www.redbooks.ibm.com>.
- 2 IBM Corporation. IBM WebSphere Enterprise Service Bus.[2009-12-04].<http://www-128.ibm.com/developerworks/cn/websphere>.
- 3 BEA Corporation.BEA aqua logic services bus. [2009-08-10].http://www.bea.com/content/news_event/s/white_papers/BEA_AQL_ServiceBus_wp.pdf, 2009.
- 4 IONA Corporation.The IONA XMLBus.[2009-09-15]. <http://www.xmlbus.com>, 2002.
- 5 李鹏. SOA 企业架构中的 ESB[硕士学位论文].西安:西安电子科技大学, 2008.21-22.
- 6 吴恒.基于 SOA 的动态路由机制的研究与实现[硕士学位论文].长沙:国防科学技术大学, 2007.10-12.
- 7 慈鑫. SOA 软件架构及其在分布式 Web 系统中的应用研究[硕士学位论文].哈尔滨:哈尔滨工业大学, 2009.4-7.