

BPEL 服务组合优化架构^①

邓 晶¹ 夏润亮² 王志坚² (1. 南京工程学院 计算机工程学院 江苏 南京 211167; 2. 河海大学 计算机学院 江苏 南京 211124)

摘 要: 针对当前服务组合存在的问题,提出了一个基于 BPEL 的服务组合优化框架。框架主要包括分布式 UDDI 服务注册中心, BPEL 设计器, 服务组合解析模块, 服务组合优化模块等。从平台框架的提出设计到平台的实现贯彻了 SOA 的设计思想, 优化服务组合, 提高运行效率。

关键词: BPEL; 服务组合; 响应时间优化; 路径优化; 优化策略

Optimization of Services Composition Based on BPEL

DENG Jing¹, XIA Run-Liang², WANG Zhi-Jian²

(1. Dept. of Computer Engineering, Nanjing Institute of Technology, Nanjing 211167, China; 2. Computer College, Hehai University, Nanjing 211124, China)

Abstract: In view of the current service combination existence's problem, proposed one based on the BPEL service combination optimization construction. The construction mainly includes the distributional UDDI service registration center, the BPEL designer, service combination analysis module, service combination optimization module and so on. From the proposition and design of the platform construction to the platform realization, it has implemented the SOA design concept, optimized service combination, raised the operating efficiency.

Keywords: BPEL; service combination; response time optimization; path optimization; optimization strategy

1 引言

随着 Internet 的不断发展,越来越多的 Web 信息项目投入使用。这些项目在解决单一问题,服务于单一部门时可以很好的解决小范围内的专业需要,但是站在全局的角度,很少考虑其它应用和整合集成的需要。单个 Web 服务难以满足实际应用的需求,如何组合已有的服务,形成新的服务组合,已成为行业信息化建设的关键。另外,在大型的建设项目中,很多 Web 服务的实时性需求很高。在高流量访问时,程序的运行效率可能会因为消息的多次转发、服务的冗长调用而降低^[1]。如何优化服务组合结构,提高服务质量,保证分布式环境中的运行效率,是我们解决的问题。为此,我们提出了一个基于 BPEL 的服务组合优化框架。在这个框架中,通过 UDDI 中心来注册发布已完成的 Web 服务;通过 BPEL 服务组合流程来整合资源,

复用原有功能模块;通过分布式优化模块来完成服务组合优化。

2 相关知识及背景

2.1 业务流程执行语言 BPEL

BPEL(Business Process Execution Language)是业界在以 XML、Web 服务为基础的诸多规范之上提出的一种业务流程定义语言,是 SOA 的关键技术^[2]。它以业务流程及其参与者的交互为基础定义了业务流程的描述语法,用于业务流程建模。BPEL 所定义的流程实质上是对一系列单个无状态 Web 服务的调用与编排组合,使得这些服务按照既定的规则进行^[3]。它只描述业务流程组合关系,不涉及业务流程实现。BPEL 流程(Web 服务组合)本身可以由 WSDL 描述的接口声明为 Web 服务。

① 收稿时间:2010-04-09;收到修改稿时间:2010-05-11

服务的重用性: 由于 BPEL 流程的调用对象是服务。一个服务在被一个流程调用的同时也可向其他流程、其他客户提供服务。同时 BPEL 流程本身也可以封装成流程向客户提供服务或是作为子流程为其他流程所重用。这种服务的可重用性为服务组合提供了强大的灵活性, 同时也提高了系统维护效率。

高度的敏捷性: 现代企业业务需求的快速改变使得相应的业务流程必须能够容易的、快速的甚至是动态的改变。一旦流程出现任何改变, 我们将不得不相应修改代码, 重新编译部署, 这样必将付出较大代价。正是由于 BPEL 可以对业务流程快速分配组合而无需重建业务逻辑, 并具有高度的松耦合性和可重用性, 才使应用系统具有高度的敏捷性。

2.2 服务组合优化

通过 BPEL 设计器, 我们得到一个抽象的服务组合 workflow。服务组合中的每个服务都有不同的副本部署在不同的服务节点上, 服务组合中的服务选择问题就成为影响服务质量 QoS 的关键。在保证负载平衡的前提下, 选择合适的服务发布节点, 使服务尽可能满足服务时间约束或者能连续使用部署在同一节点上的服务资源, 以缩短服务组合响应时间、节约服务组合路径, 达到优化服务组合的目的。

3 服务组合优化框架设计

3.1 组合框架结构

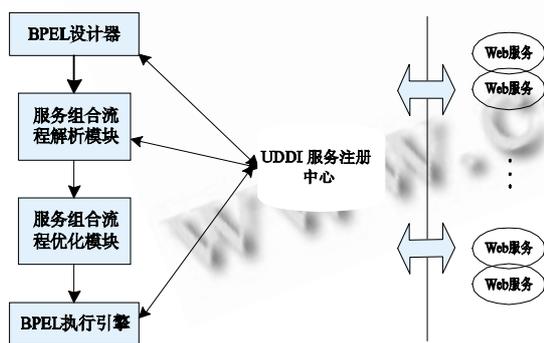


图 1 服务组合优化框架结构图

设计的核心组件如下:

① 可扩展的 UDDI 服务中心: UDDI 服务中心的任务是实现 Web 服务的发布、描述和查询。我们增加了服务的自定义拓展, 可以记录服务的副本数, 发布

的节点地址和服务的总节点数。

② 服务解析模块: BPEL 设计器将服务组合流程设计完成后, 模块首先解析组合服务所需子服务的描述文档, 获取服务发布所在的定义节点, 得到相应的服务组合参数和子服务参数。

③ 服务组合优化模块: 通过提高服务组合的服务质量 QoS 和优化改进服务组合路径来得到最终组合服务的整体最佳质量。在满足用户组合服务功能需求的同时, 通过对服务链(子服务队列)节点的优化选择, 实现需求服务的优化。主要包括服务链响应时间算法和路径选择算法。

3.2 可扩展的 UDDI 注册中心

服务平台的注册发布模块是一个行业内部私有的服务注册中心, 它提供了一个良好的交互界面与服务提供者、服务使用者(BPEL 服务执行引擎)、系统管理员进行交互。我们在开发设计时, 首先将组件开发成 Web 服务。Web 服务描述语言(WSDL)将 Web 服务描述为一组对消息进行操作的网络节点, 使用 JUDDI 来完成服务注册^[4]。JUDDI 是一个纯 Java 的 Web 应用, 因此可以配置在任何应用服务器或 Servlet 引擎中。JUDDI 由一个核心的请求处理器构成, 它用于散集接收 UDDI 请求, 调用相应的 UDDI 函数并列集 UDDI 响应。我们使用 MySQL 作为注册中心服务储存库, TOMCAT5.5 作为实现发布的服务器。因为平台在 BPEL 流程设计完成后要考虑服务优化的问题, 所以我们在服务注册时就要考虑服务节点和服务副本的问题。我们通过增加一个服务自定义单元来增加对注册服务副本数和节点数等的描述。可扩展核心代码如下:

```
// 获取 CategoryBag 自定义模块数据
BEDetail bed = new BEDetail();
CategoryBag cateBag = bed.getCategoryBag(this._strCBKN, this._styV);
// _MPraser.getCategoryBag();
service.setCategoryBag(cateBag);
Vector<BusinessService> vtService = new Vector<BusinessService>();
vtService.add(service);
```

我们可以理解为具体的服务向注册中心注册后, 将其服务的名称、服务对应的 KEY 值、服务描述、服务发布的地址和服务自定义的属性(如节点, 副本)等储

存在注册中心,并保存在 JUDDI 数据库中。在服务组合流程完成后,对外通过 QueryService, Saveservice DeleteService 来完成对服务的查询、修改和删除。

3.3 服务组合解析模块

模块依据 BPEL 标准对用户所提供的 Web 服务描述文档进行解析,提取相关的属性值,得出所有链接服务的顺序。首先,将 WSDL 描述文档提交给解析模块,文档解析模块先将 WSDL 描述文档读入,得到所有的子服务列表。然后根据 JUDDI 中查询服务的 API,查询刚才得到的每一个子服务,获取该服务流程中所有子服务总数 Service_Num,每个子服务服务副本总数 Service_Copy。解析过程如图 2 所示:

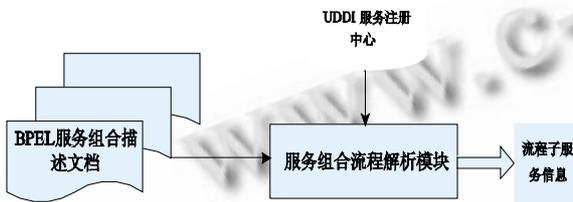


图 2 服务解析流程示意图

然后调用 JUDDI 查询服务的 API,根据子服务的名称,查找到每个子服务的部署节点序列和所有服务部署节点总数。通过服务名称对应的 BusinessKey 来唯一标识子服务。子服务的部署节点定义在 CategoryBag 中,根据条件查询 JUDDI 数据库,得出同一个 BPEL 服务名不同部署节点序列(服务链)。

```
BusinessService bs = (BusinessService) vtService.get(0);
```

```
// BusinessService 作为服务查询的标识
```

```
String strbk = bs.getBusinessKey();
```

```
strDesc = bs.getDefaultDescriptionString();//通过自定义的 API 获取描述信息
```

```
CategoryBag categoryBag = bet.getCategoryBag();//获取自定义信息
```

3.4 服务组合优化模块

在基于 intranet 的企业级应用或嵌入式环境中,服务分布是密集的,每个服务节点将部署多个服务,单个服务也将有不同的副本部署在不同节点。因此,对于一个 BPEL 服务流程会有多个符合 BPEL 序列的服

务链。如何选择合适的服务链,使得服务组合效率最优是提高应用系统执行效率的关键。我们从两方面考虑服务组合优化问题。其一,对于节点集中型网络,由于可以忽略节点间距离,主要考虑节点上服务的响应时间,以缩短响应时间作为服务链选择的优化策略。其二,对于节点之间距离较远的大范围网络,因为网络传输对系统性能影响较大,所以侧重于组合路径优化,使得在保证服务功能的基础上,优先选择部署在同一节点的服务。

3.4.1 响应时间优化

响应时间是指从提交服务请求到获得服务响应所花的时间。QoS(Quality of Service)服务质量是用来解决网络延迟和阻塞等问题的一种技术。当网络过载或拥塞时,QoS 能确保重要业务不受延迟或丢弃,同时保证网络的高效运行。在 QoS 的各种因素中,响应时间常常是最重要而且是决定性的因素。由于环境的动态性和服务本身的不稳定性等原因,使得响应时间具有难以预测的特点^[5]。另外,服务链的控制流中可能包含循环、并发等结构,使得其响应时间也变得更加不确定,从而影响了服务组合的服务质量。服务链中响应时间之和最长的路径称为关键路径。关键路径上的任何一个服务都是关键服务,其中任何一个服务的延迟都会导致整个服务链响应时间的延迟。若缩短关键服务的响应时间,会缩短服务链的响应时间;反之,则会延长服务链的响应时间。因此,顾及关键路径响应时间的服务优化组合算法的基本思想是服务链中找出关键路径,优先保证关键服务的响应时间,在不影响服务链整体响应时间的条件下,可以将响应时间短的服务资源优先配置给关键服务,以达到优化服务配置的目的。算法的实现思路为:当服务链的响应时间不能满足响应时间约束时,重新选择响应时间更快的候选服务替换现有关键服务。候选服务是提供相同或相似功能但响应时间不同的服务。候选服务的选择准则为:对于每一个关键服务,如果存在至少一个响应时间小于当前服务实例的候选服务,则选择其中响应时间最大的一个。否则,保持原来的。

3.4.2 组合路径优化

在保证基本功能和负载均衡的基础上,选择相邻节点或同一节点作为 BPEL 流程中下一个服务的提供者,使得服务组合路径最短化、最优化^[6]。当一个服务组合的服务 S_i 在节点上处理完毕时,系统必须

为 S_{i+1} 寻找一个负载较轻的匹配节点(部署了服务 S_{i+1}) 作为服务链上的后继节点。为了缩短组合路径, 引入稳定参数 $k(\text{keep})$ 来调整后继节点的选择, 使得当节点向后匹配时(节点 i 部署了后继服务 S_{i+1}), 能在不破坏负载平衡的前提下, 选择此节点为后继节点。一般来说, 后继节点的选择原则总是趋向于选择负载较轻的节点, k 通过调节后续节点的负载来对后继节点的选择产生作用。稳定参数 k 仅在节点向后匹配时才有效, 我们又引入了匹配次数参数 $t(\text{time})$ 来调整向后匹配次数, 使得在节点向后不匹配或者节点保持失败(即使节点向后匹配, 由于其它节点负载小于此节点负载, 仍选择其它节点作为服务链的后继节点)的时候, 能够选择具有最多后继匹配次数的节点作为后继节点。在 k 的配合下, 服务组合路径长度就可能越小。

组合路径优化流程如下:

① 将服务组合流程的解析数据导入优化模块。

② 判断是否存在并行。如果存在并行, 计算并行节点内匹配次数参数与路径长度差异和负载差异程度的关系。如无并行直接计算服务组合各种情况下不同节点的负载值。

③ 计算设定不同稳定参数下路径长度差异和负载差异程度。

④ 计算不同匹配次数参数下向后匹配差异程度。

⑤ 通过循环计算得出该服务组合中各种取值下的稳定参数、匹配次数参数与路径长度差异和负载差异程度的关系。

⑥ 根据计算结果提出优化策略。

3.5 组合路径优化策略提出

根据实验计算得出稳定参数 k 与匹配次数参数 t 对服务组合路径长度和负载平衡差异程度的影响。图 3.4 给出了服务组合路径长度和负载平衡差异程度随 k 与 t 在参数空间变化的等高图, 横坐标为 k 值, 纵坐标为 t 值。

实验表明随 k 和 t 的增大, 服务组合路径长度总是趋于优化。然而负载平衡性能仅在 k 和 t 取值较小的时候得到优化。根据对负载平衡的影响, 一般而言, k 和 t 的取值存在三种策略:

1) 以优化负载平衡为主, 优化服务组合路径长度为辅, 可取图 3.4 中负载平衡优化区域最低点的 k 和 t 值;

2) 以优化服务组合路径长度为主, 但对负载平衡

基本不产生影响, 则可取图 3.4 中零等高线上的点的 k 和 t 值;

3) 以优化服务组合路径长度为主, 允许对负载平衡在一定范围内产生负面影响, 如负载距离差异程度不大于 10%, 则可取图 3 中 10 等高线上(或最近)的点的 k 和 t 值。

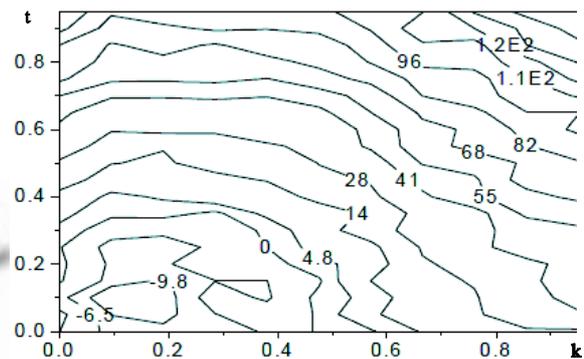


图 3 参数优化效果等高图

4 结束语

基于 BPEL 的 Web 服务组合优化是构建高效率信息系统的核心。我们以一个服务组合优化架构的研究和实现入手, 提出了实现服务组合优化的框架结构和优化策略及方法, 对不同环境和条件及不同需求下的优化做了进一步的探讨和分析。

参考文献

- 1 RayLai. J2EE 平台 WebServices. 北京: 电子工业出版社, 2005.6:29-30.
- 2 范新灿, 徐人凤, 聂哲. SOA 架构在 Web 扩展服务的探讨. 现代计算机, 2005, 12:31-36.
- 3 王莉, 刘厚泉. 基于 BPEL 的业务流程管理系统架构的研究与应用. 计算机工程与设计, 2006, 27(18):3507-3510.
- 4 饶元, 冯博琴, 李尊朝. 基于 Websvriecs 的服务合成技术研究综述. 系统工程与电子技术, 2005, 8(27):27-33.
- 5 曹利培, 刘静, 缪准扣. 基于图的 Web 服务组合优化的研究. 计算机科学, 2007, 34(2):95-99.
- 6 陈香兰, 李曦, 龚育昌. 服务组合中一种服务组合路径优化方法研究. 小型微型计算机系统, 2007, 6:1183-1185.