第 36 卷 第 8 期 计 算 机 工 程 2010 年 4 月
Vol.36 No.8 Computer Engineering April 2010
·开发研究与设计技术· 文章编号：1000—3428(2010)08—0261—03 文献标识码：A 中图分类号：TP311.5

基于 SOA 总线的资源协同组织机制

温 睿，李国益，王 飞，马亚平
(国防大学信息作战与指挥训练教研部，北京 100091)

摘 要：面向服务架构(SOA)是一种架构模型和开发概念，对总线的资源组织方式进行设计。通过总线资源的协同组织机制，实现总线上服务的动态加入和退出机制、服务状态动态变化机制、负载感知的自适应机制和节点的容错机制。在仿真与开发中，证实了该机制效率高、收敛快、适应性好，适用于大型分布式应用集成系统。

关键词：面向服务架构；服务资源；协同组织机制；总线；分布式应用

Resource Cooperative Organized Mechanism Based on SOA Bus

WEN Rui, LI Guo-yi, WANG Fei, MA Ya-ping
(Department of Information Operation & Command Training, National Defense University, Beijing 100091)

【Abstract】This paper discusses some concepts on Service-oriented Architecture(SOA), insists that SOA is a developing methodology rather than a specific technology. Moreover, the paper designs the service cooperative mechanism based on SOA. Many functions are supported, including service dynamics, the load-aware self-adaptation and fault tolerance. According to simulations and applications, this mechanism has low bandwidth, fast convergence and the ability to recover from emergencies quickly.

【Key words】Service-oriented Architecture(SOA); service resources; cooperative organized mechanism; bus; distributed application

1 概述

SOA 是一种软件构架，是一种粗粒度、松耦合的架构思想，它由服务和基础设施构成，通过运行于基础设施之上的服务和服务的聚合灵活地实现功能需求[1]。现在的研究已经对 SOA 进行了详尽的论述，但仍有不少问题没有解决，最重要的就是服务资源的协同组织，特别是总线服务资源的发现与动态同步问题并未设计。本文借鉴网络路由发现算法和网格资源计算方法[2-3]，论述了 ESB 组成结构和自组织机制。

2 总线拓扑结构

要实现信息自动同步，以及全局域的系统自组织，必然要采用基于域内服务注册的统一基础设施，如图 1 所示，这是总线设计的高级层次。

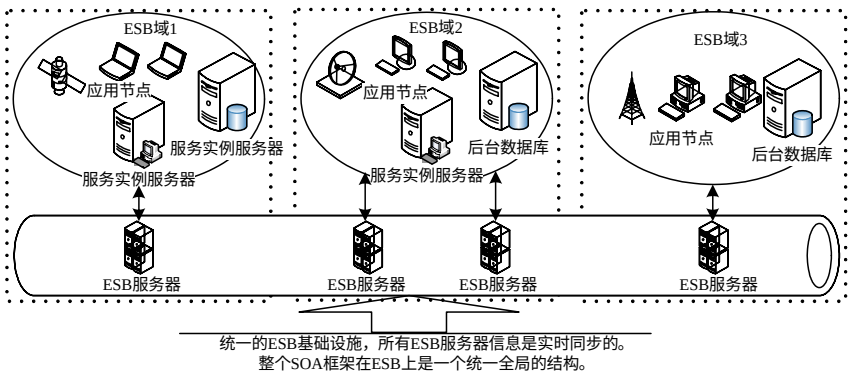

图 1 ESB 总线基础设施示意图

3 协同组织机制的概念与信息结构

3.1 协同组织机制的概念

ESB 是构建在底层互联互通的物理网络之上的一层概念总线，通过 ESB 的实施将底层各种物理网络透明化，通过服务资源的自组织机制来完成 ESB 信息的建立和维护过程，具体包括初始化、自主演化、服务实例资源的动态性维护以及容错机制 4 个方面。协同组织机制可自行适应服务的更新和拓扑结构的变化，能自行根据各服务实例副本的忙闲程度进行负载均衡，有利于改善体系的性能。同时，机制的收敛时间，即采用算法的所有节点在拓扑发生变化后所有信息达成一致的时间，应当较短，能够迅速地收敛是效率的体现，也能够快速部署更新服务。

3.2 信息结构

采用注册机制，任何 ESB-S 或实例启动后都要向总线进行注册。主要采用基于属性的方法对各类资源进行描述，通过一组基于<名,值>对的属性来描述每个服务资源的信息。例如，通过属性集合 U={*CPU*=2 GHz, *Mem*=2.0 GB, *Disk*=160 GB,…,*Mem_Load*=1 024 MB}来描述服务器性能，如果属性值为非数值类型值，则可通过哈希函数等数学手段进行数值化处理。在实际运行中，服务在真实计算环境下的动态属性更为重要，这个动态属性不仅包括了服务器处理信息的能力，同时也包括了网络的传输和处理能力，动态信息表示了资源的实际可用能力，是系统调度实施正确调度和资源选取决策的基础。

根据总线运行主特征，注册信息中应包括如下部分：(1)*register_taddress*，地址，表示 ESB-S 和服务实例的地址，使用网络

基金项目：国防大学科研计划立项课题基金资助项目(09GDA030)；海军装备预研基金资助项目

作者简介：温 睿(1982－)，男，工程师，主研方向：作战建模与仿真，军事运筹；李国益、王 飞，硕士；马亚平，教授、博士生导师

收稿日期：2009-11-18 **E-mail：**wenrou6274219@163.com

层的 Ipv4 地址与 Mac 地址共同表示。(2)*register_type*，注册类型，表示注册的类型，分为 ESB-S 注册、服务实例和消费者端注册 3 种类型。(3)*service_id*，服务编号，当注册类型为服务实例时，这里为服务编号；当注册类型为 ESB-S 时，为 0；服务编号根据服务全局统一，某一特定的服务有全局统一的服务编号，利于服务注册与版本更新。(4)*area_id*，所在 ESB 域号，用于区别不同的 ESB 域，ESB-S 统一进行域的设置；服务实例与应用客户端只向本域的 ESB-S 进行注册，其 *area_id* 设置为 0，当 ESB-S 向邻居 ESB-S 发布该服务实例时则设置为本 ESB 域号。(5)*net_delay*，网络时延，直观地表现出网络传输和服务处理的时延。(6)*register_time*，已注册的时间，初始注册时为 0，随注册时间递增，时间越少优先级别越小，就有注册不断被更新，以此来测定版本更新、服务超时等。(7)*load_status*，ESB-S 或服务实例负载情况，如果超过某个阈值就应当进行负载均衡，避免过高负载的情况。(8)*current_status*，当前注册状态，注册条目分为有效状态、无效状态与检验状态，用以确保服务的正确性。(9)*if_expanded*，是否具有扩展参数，为以后 SOM 拓展留下接口。

3.3 报文类型设置

主要有以下几种报文形式：(1)Hello 报文，用于 ESB 邻居的发现和恢复。所有 ESB-S 都是邻居，都应当建立起邻接关系。(2)ACKs 报文，Hello 报文与更新报文的应答报文。ACKs 报文中包含的时间 *time* 为 Hello 报文和 ACKs 报文往返时间之和，$\left[(time_{Hello}+time_{ACKs})/2+net\_delay\right]/2$ 即为报文往返处理时间 net_delay，这使 net_delay 能根据网络状况进行实时更新。(3)Update 更新报文，在 ESB 信息已经同步后，应当随时分发全局 ESB 服务状态或者本域服务状态。更新报文的发送机制采用事件触发和定时发送 2 种，当有新的服务需要进行更新时，对每个 ESB-S 发送服务更新；或者到达更新阈值时，定期发送本域的注册服务。(4)Request 请求和 Reply 答复报文，用于 ESB-S 启动时主动进行信息同步。在与不同域 ESB-S 同步信息时，只接受与邻居 ESB-S 同域服务实例，确保该域服务的可访问性。

4 协同组织机制设计

4.1 初始化

在 ESB-S 初次启动时，采用广播或者组播的方式发送 Hello 报文，和某一 ESB-S 建立起邻接关系后，同步已有 ESB 全局信息，并向所获得的其余 ESB-S 发送 Hello 报文，如图 2 所示。

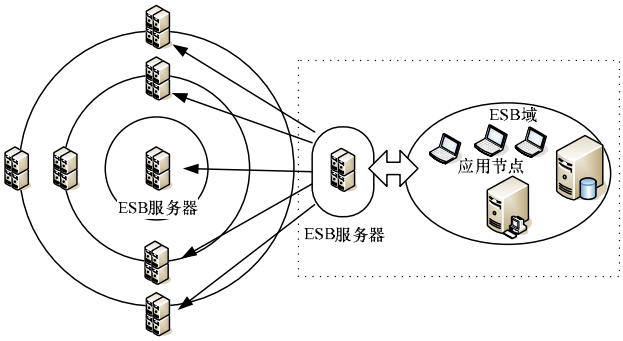

图 2 自组织状态

一次启动之后，即将 ESB-S 信息记录下来，置 *current_status* 为无效状态，再次启动时，使用 Hello 报文逐次验证原有邻居 ESB-S，如果都失效，则再次采用广播或者组播的方式发现邻居，直至确认自身为初始启动。

4.2 负载的自感知

为保证 ESB 的自组织性和高效率，采用了一种简单的自适应机制。其关键就是如何定义衡量一个 ESB-S 负载 l 的指标。由于描述资源的信息是一些<名,值>对，管理服务资源对存储空间的要求不大，其开销主要体现为计算能力和网络传输能力的使用，因此，可以采用以 ESB-S 本身 CPU 占用量 *cpu_load* 和网络传输时延 *net_delay* 作为参数，用 $\langle cpu\_load, net\_load\rangle$ 较为全面地反映服务处理能力，具体 l 的计算方法为：$l=(cpu\_load+net\_able)/2$。

4.3 邻居确认与重传机制

邻居的无效及启动虽然可通过 *register_time* 的超时来检测，可是应当有一个确认与重传机制来防止意外产生的损失。

4.3.1 邻居的发现

在 Hello 报文发送出去后，只有从邻居 ESB-S 那里收到 ACKs 报文才能建立起正式邻接关系。在邻接关系已经建立后，为确保关系的维持，经过一个 *hello_interval* 的时间间隔，就应该重发 Hello 报文。下层物理网络传输介质的不同，网络传输的时间也是不同的，例如光纤网、以太网和帧中继、ATM 网络的链路速率肯定是不一样的，那么，在 *hello_interval* 值的确定上就应该考虑到这种综合因素，采用 $hello\_interval=max(n\times net\_delay, designed\_timer)$，则能够较好地考虑网络特定传输因素。这里 *designed_timer* 为一指定确定值，因为在局域网中 *net_delay* 非常小，为确保网络不致拥塞而设置。

4.3.2 邻居的无效

如果一个 Hello 报文通过单播方式发送出去，经过一个 *hello_interval* 时间而没有从邻居那里收到 ACKs 报文，那么就在每个 *net_delay* 间隔时间向该邻居发送 Hello 报文，经过特定的 m 次单播重传还没有收到一个 ACKs 应答，便设定该邻居 *current_status* 为无效，这样检测失效的时间应该为 $m\times net\_delay+hello\_interval$。

4.3.3 邻居的恢复

在因意外宕掉的 ESB-S 重新启动后，会主动发送 Hello 报文请求建立邻接关系，在发送 ACKs 后，便可置 *current_status* 有效，恢复邻接关系。

4.4 协同组织机制的状态变化

在协同组织机制中，最关键的是一套规则，使 ESB 和服务信息能够自动组织、分发、应用，用于说明整个体系是如何自行运转的，如图 3 所示。

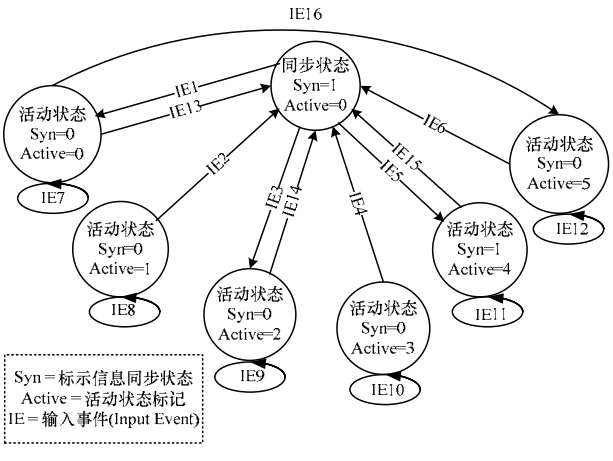

图 3 协同组织机制状态变化图

4.4.1　状态变化的输入事件

当 ESB 信息进入全局同步状态时，总线处于稳定状态。在产生输入事件的任何时候，都会触发重新进行信息的交换同步。其输入事件如下：

IE1，本 ESB-S 启动和注销。

IE2，在本域产生新的服务实例、发现某 ESB-S 或链路异常时向总线发送 Update 报文进行信息同步。

IE3 到 IE6 分别是收到一个 Hello, Update, Request 和 Reply 报文的事件。

IE7，除了产生自身启动和注销的事件。

IE8，除了有新服务实例产生或某 ESB-S 不可达的事件。

IE9 到 IE11 分别是除收到 Hello, Update 和 Request 报文以外的事件。

IE12，没有收到答复 Reply 报文。

IE13，向其余服务器发送的 Hello 报文确认全部返回。

IE14，发送 ACKs 报文。

IE15，发送完回应报文。

IE16，发送 Request 报文，请求信息同步。

4.4.2　状态变化类型

由于有多种类型的输入事件源引起状态的改变，因此处于 Active 的状态时就说明可能发生了一些类型的输入事件。采用由信息同步状态 *Syn* 和活动状态标记 *Active* 组成的状态标识 $\langle Syn, Active\rangle$ 来描述 ESB-S 所处的状态。

(1) $\langle 1,0\rangle$，ESB 信息全局同步状态。

(2) $\langle 0,0\rangle$，向所有邻居 ESB-S 发送 Hello 报文。

(3) $\langle 0,1\rangle$，内部注册注销，向邻居发送 Update 报文。

(4) $\langle 0,2\rangle$，改变本地邻接关系。

(5) $\langle 0,3\rangle$，改变本地服务注册状态，新增或置无效。

(6) $\langle 1,4\rangle$，建立本域服务状态准备发送。

(7) $\langle 0,5\rangle$，同步信息，新增外域服务状态信息。

5　仿真研究

ESB 基础设施实际上是一个庞大的分布式系统，协同组织机制在实施之前应当进行精确的建模与仿真，用以进行性能的评估。利用 OPNET 仿真工具，通过事件驱动方式，仿真了具有 100 个 ESB-S 节点的总线结构，每个 ESB-S 平均负载 25 个~35 个服务实例资源，底层网络采用 RIP 网络，消费者查询请求的发生采用泊松随机算法生成，并采用随机过程进行退出。模拟了复杂互联网络中协同自治机制的一些特性，如图 4 所示，主要是信息的收敛时间，以及数据占总吞吐量比例，仿真发现占用带宽少，适应性好。

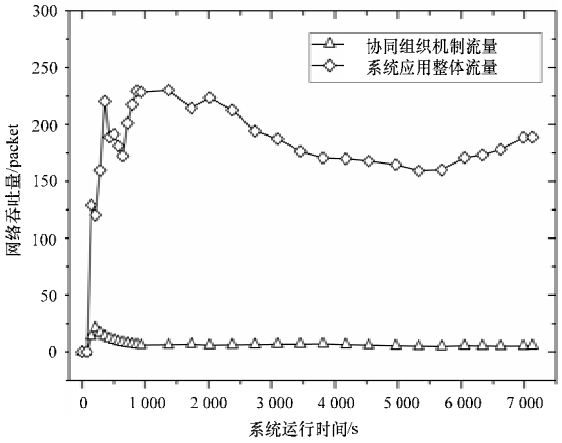

图 4　系统运行时网络流量示意图

6　结束语

本文将采用资源协同组织机制的总线应用于“联合作战训练信息系统”的开发改造，取得很好效果。该系统在构件库的基础上建立 SOA 的服务总线，实现第 3 个层面上即应用层面的可组构。依托现有网络基础设施构建大型分布式系统，应充分考虑底层的异构性与复杂性。在网络层上构建逻辑的总线层，可以实现全维作战域内广域网节点的互联互通互操作，和大规模数据的订购分发应用[4-5]，使全维数据资源处于随时待命状态。所开发的构件以服务的形式按照协同组织机制进行集成，即建立服务总线，然后根据仿真运行的需要通过作战行动流引擎驱动相应服务，完成作战过程流的调用。

参考文献

[1] 麻志毅, 陈泓婕. 一种面向服务的体系结构参考模型[J]. 计算机学报, 2006, 29(7): 1011-1019.

[2] 李　静, 陈蜀宇, 田　东. 一种网格环境下的动态负载均衡机制[J]. 计算机工程, 2008, 34(4): 19-21.

[3] 田国忠, 于　炯, 侯　勇, 等. 基于资源有效度的网格工作流任务调度算法[J]. 计算机工程, 2008, 34(11): 80-82.

[4] 易峥荣, 卜　炜, 葛序风, 等. 基于 SOA 的数据协同模型[J]. 计算机工程, 2009, 35(4): 261-264.

[5] 刘思中, 曹　健. 基于 SOA 的事件协同感知模型[J]. 计算机工程, 2009, 35(3): 48-50.

编辑　任吉慧

(上接第 260 页)

参考文献

[1] 张　兵, 吴泉源, 彭　坤. JCA 中资源适配器的设计和实现[J]. 计算机工程, 2006, 32(23): 1-2.

[2] Sun Microsystems. J2EE Connector Specification V1.5[EB/OL]. (2003-10-21). http://whitepapers.zdnet.com/Software+and+Web+Development/Internet+and+Web/J2EE/.

[3] IBM. WebSphere Adapter Development Redbook[Z]. (2006-03-20). http://www.redbooks.ibm.com/redpieces/abstracts/sg246387.html.

[4] Maheshwari P, Kim Ji-Ho. Analysing Resuability Aspects in Java Connector Architecture[C]//Proc. of the 11th APSE'04. Busan, Korea: [s. n.], 2004: 2-3.

[5] 张建锋. 使用JCA Inbound 实现信息流入集成[J]. 计算机与信息技术, 2006, 13(7): 68-70.

[6] 雷爱平, 尹建伟, 陈　刚, 等. 面向 EAI 的高可用性松耦合的扩展 JCA 架构[J]. 计算机应用, 2005, 25(2): 1-2.

编辑　索书志