面向分布式控制系统的实时SOA研究与应用

杜立新[1,2] 刘士军[1]

(山东大学计算机科学与技术学院 济南 250101)[1] (济南大学信息科学与工程学院 济南 250022)[2]

摘 要 SOA(Service Oriented Architecture)已广泛应用于传统信息系统领域。如何将SOA思想应用于社会上广泛存在的分布式控制系统的构建,是服务计算领域的一个新的研究课题。针对分布式控制系统的特点及面向服务的思想,提出一种可有效解决消息可靠性传输、服务实时性处理的层次化实时SOA模型,给出各层次模型的详细思路与实现机制。基于该模型,设计并实现了电力配网自动化系统的底层多网络冗余实时消息总线原型系统,并给出了相关结论。

关键词 面向服务架构,实时分布式控制,消息总线,多网冗余

中图法分类号 TP302.1 **文献标识码** A

Research on Real-time SOA towards Distributed Control Systems

DU Li-xin[1,2] LIU Shi-jun[1]

(School of Computer Science and Technology, Shandong University, Jinan 250101, China)[1]

(School of Information Science and Engineering, University of Jinan, Jinan 250022, China)[2]

Abstract SOA (Service Oriented Architecture) has been used wildly in traditional information system area. But how to construct a distributed control system widespread used in our society with the novel idea SOA is a new issue in service computing research area. With the characteristics of distributed control system and SOA, we proposed a layered real-time SOA model which could improve message transport reliability and real-time service processing. We discussed its detailed mechanism on message model, message transportation and service processing. Based on this model, we developed a multi net redundant real-time message bus prototype system for power distributed automation system and gave our results.

Keywords SOA, Real-time distributed control, Message bus, Multi net redundancy

1 研究背景

分布式实时控制系统在当前社会中普遍存在,其主要特点是通过可靠的通讯网络实现对物理位置分布广泛的传感器、控制器等设备的数据采集分析及远程遥控、遥测、遥调、遥信等操作。比较典型的有电力配网自动化、调度自动化等系统。此类系统在可靠性、实时性及容错能力方面较传统应用有更高的要求。过去分布式控制系统软件一般都是采取封闭定制模式,即根据特定应用需求来设计软件,产品很难实现复用,后期的维护升级工作困难。另外,随着物联网的提出与发展,人们意识到大量分布式控制系统、数据采集系统是构成物联网的基础应用之一,必须将各类控制、数据采集系统、信息管理系统、公共服务系统等的数据集成[1]并统一为物联网应用服务。此时传统的定制模式已经成为制约物联网软件应用构建的瓶颈。另一方面,面向服务的架构(SOA)已经成为事实上构建新型企业应用的主要指导思想,其粗粒度、松耦合的特点也使得分布式应用的构建更加简单。因此,研究SOA在分布式实时控制应用环境下的特征、变化及应用模式成为了当前面向服务领域研究的一个新的课题,即通常意义上的实时SOA研究。

2 实时SOA研究现状

当前针对实时SOA的研究还处于起步阶段,国内相关研究较少,国外的研究成果大部分集中在实时SOA架构方面。Cucinotta等针对工业控制系统给出了一个增强了实时性的SOA架构[2],该架构本质上是一个面向QoS的通讯中间件,主要包括协议层和API层两个层次。Tsai等描述了一个概括性的实时SOA框架[3],并对实时环境下的服务发现、组合、发布、执行等进行了初步探讨,指出实时性问题是实时SOA必须解决的基本问题,但文中未给出保障实时性的具体机制与算法。Luis提出了一个实时消息中间件架构[4],指出实时ESB需包括网络通讯、消息调度及接口3个层次,用以满足实时消息的可靠传输。Lin等通过一系列的文献[5-7]说明了其所研究的RT-Lama实时SOA项目的详细架构,以及为了提

到稿日期:2011-05-27 返修日期:2011-07-15 本文受国家高技术研究发展计划(863)(2009AA043506),山东省自然科学基金(ZR2009GM028)资助。

杜立新(1972—),男,博士生,讲师,主要研究方向为服务计算,E-mail:du.lixin@163.com;刘士军(1972—),男,博士,副教授,主要研究方向为服务计算、云计算。

高服务的实时性所提出的服务预约机制及相关算法。Pohl等对SOA在分布式控制系统中的应用进行了研究[8]，将每个传感器/控制器的功能封装到一个控制服务中，这些控制服务可以组成层次结构为更高层次的应用提供服务。胡侃等提出了一种基于事件驱动的针对传感器网络实时服务的中间件框架[9]。张珂等提出一个基于SIP的异构实时服务集成平台架构[10]，重点描述了该架构下的服务发现、寻址等机制，对于服务之间的交互机制未给出明确说明。朱旭东等对弱硬实时系统的概念进行了说明，并提出一个实现平滑调度的约束规范[11]。根据现有研究成果以及分布式控制系统的特点，我们认为实时SOA模型必须解决下面几个问题：

(1)高效、可靠通讯机制。

(2)实时服务的抽象与封装方法。

(3)确保服务处理的实时性、可靠性。

上面问题(2)属于软件分析与设计领域，在此不进行探讨。下面将针对问题(1)和(3)给出详细的解决方法。

3 实时SOA模型

3.1 整体架构模型

根据分布式实时控制系统的特征，我们将实时SOA架构划分为4个层次：物理层、实时消息层、实时服务层、企业应用层，如图1所示。物理层大量的智能设备、传感器、控制器、RFID等，是系统数据的来源和指令执行者。实时消息层以消息的形式完成服务与服务、服务与设备之间的通讯。实时服务层负责完成实时服务的抽象及处理(组合、迁移、容错等)。这里的实时服务包括物理设备所封装成的服务以及所抽象的独立逻辑功能。企业应用层实现基于实时服务层的服务构建的更高层次的逻辑功能处理，一般是从参与者的角度进行构建。本文重点论述实时消息层和实时服务层的详细模型与机制。

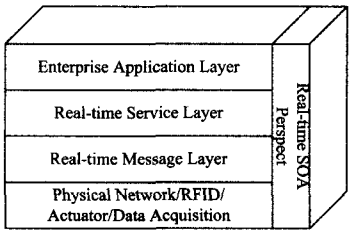

图1 实时SOA层次结构模型

3.2 实时服务层模型

实时服务层是整个实时SOA架构的核心，实现原子服务封装及服务本地执行与处理。其详细结构如图2所示。

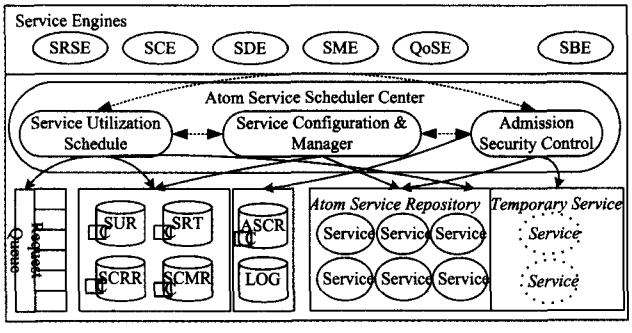

图2 基于实时SOA的服务层架构

3.2.1 服务处理引擎SE(Service Engines)

服务处理引擎由一系列可动态加入、撤除的引擎组成，这些引擎本身封装一定的逻辑功能，对外提供合约服务，具体说明如下。

1)SRSE(Service Response Scheduling Engine)：服务响应调度引擎。SRSE接受原子服务的调用请求，根据请求的优先级别、性能要求进行排队，经过原子服务调度中心调度执行。若请求超过服务提供者的服务能力，SRSE会调用预测与分析服务来满足最需要的请求者服务请求。

2)SCE(Service Composition Engine)：服务组合引擎。SCE接收服务的调用请求，根据服务请求及组合规则对原子服务进行组合，并通过ASSC调度执行。SCE可以通过SME将其它主机的服务迁移到本地组合执行，也可以通过SBE代理组合其它主机服务，或者与其它主机的SCE自主组合。

3)SDE(Service Discovery Engine)：服务注册及发现引擎。将本地服务注册到服务中心，并通过注册中心获取其它主机服务信息。

4)SME(Service Migration Engine)：服务迁移引擎。将其它主机的服务迁移到本地或协助其它主机迁移本机服务。

5)QoSE(Quality of Services Engine)：服务质量管理引擎。对本地服务的服务质量进行管理与统计，并更新服务注册中心的服务质量信息。

6)SBE(Service Broker Engine)：服务代理引擎。代理执行其它主机的服务。与SME引擎不同，SBE实现本地请求的远程代理。

3.2.2 原子服务调度中心ASSC

实时SOA中的原子服务是指对一个或多个物理设备对外展现的功能所封装成的独立最小服务。原子服务被上层软件服务看作是具有独立功能的逻辑设备。根据物理设备的特性不同，原子服务也具有不同的特性。可以并行操作的设备所对应的抽象原子服务可以提供多路并行服务，并行服务的能力具有有限的特征。不能并行操作的设备则无法提供并行服务。因此，ASSC对于服务请求者需要根据自己的服务能力进行排队和调度执行。ASSC主要包括3个调度器：

1)SUS(Service Utilization Scheduler)：服务使用调度。负责根据每个服务的服务能力来调度本地服务的执行。

2)SCM(Service Configuration Manager)：服务配置管理器。负责配置管理原子服务执行。

3)ASC(Admission Security Control)：安全准入控制。根据设备的不同类型，提供基于硬件或软件的操作安全检查、验证等。

上述各调度器统一协调操作，共同完成原子服务的调度与执行。

3.2.3 存储库

每个部署服务的主机都有多个存储库，存储每个服务运行的信息，如执行时间、执行效果、服务截止期满足情况等。所记录的信息用于对今后服务执行编排的预测分析。同时，在独立的控制中心存储库中对应存储整个服务网络的公共服务使用数据，并将各主机存储库连接成大型分布式存储库，以方便进行跨主机服务操作(如不同主机间的服务组合、迁移、调度等)。每个存储库都带有一个协同操作器，以实现不同主机间的协同计算。具体包括：

1)服务使用情况记录库(SUR,Service Utilization Repository)。记录本地原子服务的详细执行情况,并提供给 SUS 进行服务执行预测分析,以达到科学调度目的。

2)服务组合规则库(SCRR,Service Composition Rules Repository)。记录本地服务的静态和动态的组合规则,提供给 SCE 进行服务组合。

3)服务配置管理库(SCMR,Service Configuration Manger Repository)。记录本地服务的各种静态、动态配置信息。

4)服务路由表(SRT,Service Router Table)。记录其它服务的路由信息。

5)安全准入控制规则库(ASCR,Admission Security Control Rule Repository)。记录本地服务的安全准入规则、硬件设备安全操作校验、软件安全验证等信息。

3.3 消息层模型

可靠消息通讯是实时 SOA 的基础。服务之间的通讯由图 1 所示的实时消息层处理,我们称之为消息总线。一个完整的消息总线由一个中心节点和若干个服务节点组成,中心节点负责管理协调其它节点的运行,若中心节点失效,可以由其它节点选举产生新的中心节点。节点整体分为 4 部分。消息层结构如图 3 所示。

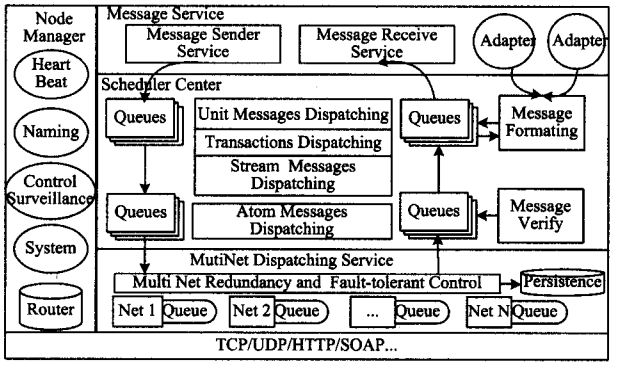

图 3 实时消息处理层架构

3.3.1 节点管理(Node Manager)

节点服务主要实现消息总线中各服务节点心跳管理、名称服务、工况及其它系统管理功能(如中心节点选举等)。中心节点起到对总线中普通服务节点的消息通讯管理与协调的作用,中心节点服务失效时由其它服务节点自动推举新的中心节点。

3.3.2 消息服务(Message Service)

消息总线以消息服务的形式对上层的服务层提供消息的发送、接收等服务。

3.3.3 消息调度中心(Message Scheduler Center)

消息调度中心是整个消息中心的核心,负责将不同类型的消息分解封装成原子消息并根据预定义的调度策略进行发送。以下将对消息模型及机制进行说明。

3.3.4 多网络分发服务(Multi Net Dispatching Service)

为了提高消息发送的可靠性与实时性,消息总线提供基于多网络传输的机制,既可以冗余传输又可以并行传输。假设总线共有 M 个网络同时通讯,系统可以按照式(1)动态调整效率与可靠性分配:

$$M=R+N \quad (0\leqslant R\leqslant M,0\leqslant N\leqslant M) \tag{1}$$

式中,R 表示冗余度,即同时用 R 的网络传输指定消息并校验消息的一致性,达到确保消息可靠传输的目的。N 表示并行度,即可以把一组消息(例如流消息拆分后的多个原子消息)分布到 N 的不同网络同时传输,相对一个网络消息提高 $N-1$ 倍。

3.3.5 消息模型

根据消息规模,我们定义了 3 种不同形式的消息:

• 原子消息。原子消息是消息总线两个节点间的一次消息传输的基本单位,具有不同的类型和优先级,由调度中心负责排队与调度。

• 单消息。是指具有固定格式的消息,是外层服务之间发送消息的基本单位。在消息总线进行传输时,单消息根据规模的大小可能被自动拆分成多个原子消息发送。

• 流消息。是指服务之间交互的不定长或超长的消息,例如数据库查询结果、二进制文件、大段内存数据等。流消息需要拆分成多个原子消息进行发送。一般来讲流消息所拆分的原子消息的级别较低,其传输可随时被暂停,以插入更高级别的原子消息传输。

实时应用系统中不仅有数据的传输,还有各种实时监控数据、控制命令等的传输。根据数据的不同特征,我们定义了如下不同消息种类,主要有:

• 控制消息。控制消息具有最高的优先级,其内部封装了针对特定设备的控制命令,总线负责可靠地将控制消息传送到目的主机的相应控制服务去执行。一般情况下,控制消息不可拆分,直接封装成原子消息发送。

• 实时数据消息。实时数据消息对各种实时监测数据进行封装,这些实时监测数据的特点是数据量大但每个数据的类型长度基本固定。实时数据消息具有低于控制消息的优先级。在总线中传输时,实时数据消息根据需要可以被拆分成多个原子消息进行传输。

• 系统消息。系统消息指的是维护整个服务总线、消息总线运行所定义的消息,其优先级根据消息指令的不同而不同。

• 应用数据消息。应用数据消息指的是除去上面 3 种消息以外的服务之间需要传递的各种数据、命令等,其优先级最低。

总线进行消息调度时,会根据消息的形式和种类及预先定义的调度策略进行调度。

3.3.6 消息事务

在分布式控制系统中,很多情况下对于物理设备的一次控制要分解成多个步骤,通过一系列的指令操作来完成。另一方面,对于发送给物理设备的指令可能要通过多个中继节点转发来执行,我们采用消息事务的思想来解决此类问题。消息事务是服务请求者和提供者之间通过多次消息交互来完成服务调用的过程,过程中可以使用第三方服务作为中介。从外部观察者的角度只观察到服务请求者和提供者的行为,内部对第三方中介服务的使用有两种情况:

• 分布式情况下需要中介中转服务。

• 事务逻辑执行过程中本身需要使用中介服务的逻辑。

消息事务过程包括申请、执行和取消 3 个步骤。假设请求者为 R,提供者为 P,中间顺序使用的第三方服务队列为 $t_1,t_2,\cdots,t_n$,消息事务处理过程算法描述如下:

Step 1 R 发送事务申请消息到 t_1。

Step 2 t_1 收到事务请求后若能满足服务请求,则继续

发送请求至 t_2，同时开始事务执行准备。若不满足，则回复拒绝，事务请求中止。本步骤依此类推执行 $t_1, t_2, \cdots, t_n$，直到 P 满足请求，转 Step 3。

Step 3　P 满足请求，消息发送到 t_n，t_n 签署同意意见后发送到 t_{n-1}。依此类推直到 R 收到所有服务的请求，队列建立。

Step 4　R 和 P 开始执行服务过程。

Step 5　服务执行结束后，R 发出服务完成消息，经过第三方队列到达 P，所有接收者标记事务完成。

Step 6　结束。

4　验证与应用

按照本文所述思路，我们设计了基于实时 SOA 的电力配网自动化主站系统，服务模型如图 4 所示。原子物理设备服务层以实时原子服务的形式对开关、刀闸、断路器、FTU 等设备及操作进行封装；核心电力逻辑服务层提供电网拓扑、线损分析、故障隔离等较高层次的实时分析与控制服务；分析与应用服务层实现企业应用相关的服务封装。上层服务通过调用、组合下层服务或本层次服务来构建更高级的服务。实时消息总线层完成服务间消息通讯，同时实现多网冗余及消息容错功能。

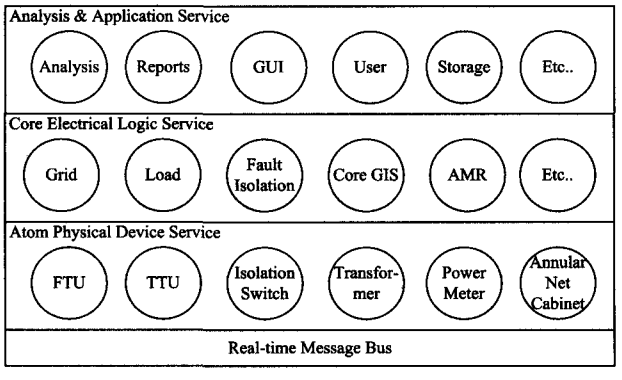

图 4　基于实时 SOA 的配网自动化系统服务模型

为了保证消息通讯的可靠性和效率，我们进行了实验验证。实验环境是 3 台普通台式机，CPU 是 Intel Q8200，内存 4G，操作系统为 SUSE Linux Enterprise Server 11。网络环境为 3 个独立百兆网络。分别在单网络、双网络、三网络环境下发送单个大小为 100M 到 900M 字节的流消息，流消息由总线自动拆分成 1k 字节大小的原子消息进行传输。发送方发送消息并收到接收方的确认消息为一次成功消息传递。实验平均结果如表 1 所列。

表 1　消息总线通信效率测试结果(单位：ms)

网络数\规模	100M	200M	300M	400M	500M
单网络	9170	18361	27758	36652	45871
双网络	4612	9186	13790	18360	23720
三网络	3100	6155	9229	12694	15935
网络数\规模	600M	700M	800M	900M	
单网络	55016	64533	73727	82980	
双网络	28499	33245	37866	42644	
三网络	19308	22490	25276	28544	

根据实验结果生成趋势图，如图 5 所示。

从表 1 和图 5 可以看出，实验总线单网络实际传输速率约 10M 字节/s，达到 100M 网络的实际传输最高速率；并行网络数和传输速度成正比，且多网络传输效率高于单网络传输。随着消息规模的增加，多网络传输效率提升更为明显。在满负荷消息传输过程中，监测 CPU 占用率一般维持在 15%左右。测试结果说明基于多网络的消息总线的架构模型是正确的，适于大数据量实时数据并行传输。

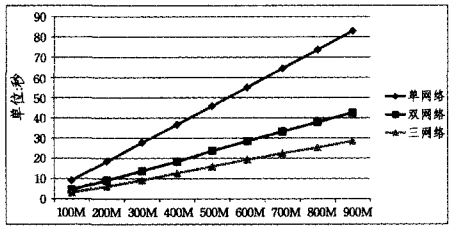

图 5　多网络消息总线传输趋势图

结束语　SOA 思想在分布式控制系统领域的应用面临着新的挑战。本文提出的实时 SOA 模型充分考虑了服务通讯与服务处理所面临的问题，适用于构建带有大量实时监测数据和控制的软件系统。如何确保本模型下的服务组合、响应等处理的实时性，是我们下一步研究的目标。

参 考 文 献

[1] Du Li-xin. SOA-based Intergration of Enterprise Vertical Applications[C]// Proc. of ICCASM 2010. IEEE Press, 2010, 12: 122-126

[2] Cucinotta T, Mancina A, Anastasi G F, et al. A Real-time Service-oriented Architecutre for Industrial Automation[J]. IEEE Transactions on Industrial Information, 2009, 5(3): 267-277

[3] Tsai W T, Lee Y H, Cao X Bn. RTSOA: Real-time Service-oriented Architecture[C]// Proc. of Second IEEE International Symposium on Service-oriented System Engineering. IEEE Computer Society Press, 2006: 49-56

[4] Luis G-E. Building an Enterprise Service Bus for Real-time SOA: A Messaging Middleware Stack[C] // Proc. of Annual IEEE International Computer Software and Applications Conference. IEEE Computer Society Conference Publishing Services, 2009: 79-84

[5] Lin K-J, Panahi M, Zhang Yue, et al. Building Accountability Middleware to Support Dependable SOA[J]. IEEE Internet Computing, 2009, 13(2): 16-25

[6] Panahi M, Nie W, Lin K-J. A Framework for Real-time Service-oriented Architecture[C]// Proc. of CEC2009. IEEE Computer Society, 2009: 460-467

[7] Panahi M, Nie Wei-ran, Lin K-J. The Design and Implementation of Service Reservations in Real-time SOA[C]// Proc. of ICEBE'2009. IEEE Computer Society, 2009: 129-136

[8] Pohl A, Krumm H, Holland F, et al. Service-orientation and Flexible Service Binding in Distributed Automation and Control Systems[C]// Proc. of AINA2008. IEEE Computer Society, 2008: 1393-1398

[9] 胡侃，刘云生，李坚. 一种保证传感器网络实时服务的中间件机制[J]. 计算机科学，2007，34(12)：5-60

[10] 张珂，黄永峰，李星. 异构实时服务集成平台的研究与实现[J]. 计算机工程，2009，35(7)：1-3

[11] 朱旭东，常会友，衣杨，等. 基于平滑调度的弱硬实时系统约束规范[J]. 计算机科学，2010，37(3)：205-207