

面向开放移动网络环境的新一代中间件技术研究

李琪林¹ 王敏毅² 周明天³

(四川电力试验研究院系统室 成都 610072)¹ (爱立信广州研发中心 广州 510665)²

(电子科技大学计算机学院 成都 610054)³

摘要 结合现有网络计算环境的特点,对适用于开放、移动网络环境的新一代中间件技术进行了研究。在分析今天网络环境关键特征的基础上,提出了新一代中间件系统需要满足的技术需求;从所针对的问题角度对新一代中间件系统面临的主要挑战进行了详细的讨论,包括上下文感知、环境自适应、开放式协作、服务发现和普适互操作等,并给出了一些解决方案。

关键词 开放性,移动性,上下文感知,适应性,开放式协作,服务发现,普适互操作

中图分类号 TP393.02 **文献标识码** A

Research on New Generation Middleware Technology in the Open and Mobile Network Environment

LI Qi-lin¹ WANG Min-yi² ZHOU Ming-tian³

(Power Department, Sichuan Electric Power Test and Research Institute, Chengdu 610072, China)¹

(Ericsson R&D in Guangzhou, Guangzhou 510665, China)²

(School of Computer Science and Engineering, University of Electronic Science and Technology, Chengdu 610054, China)³

Abstract Combining the characteristics of current network computing environment, research on new generation middleware technology which is suitable for open and mobile networking environments was presented. Based on the analysis to the key characteristics of today's network environment, new technical requirements for new generation middleware were identified. Subsequently, some key challenges such as context-awareness, environmental adaptation, open coordination, service discovery and pervasive interoperability for new generation middleware were discussed in detail. Further, some real solutions were showed as well.

Keywords Openness, Mobility, Context-awareness, Adaptability, Open coordination, Service discovery, Pervasive interoperability

上世纪 90 年代以来,由于中间件的出现及广泛采用,分布式系统的设计者已经不必花费太多的精力去考虑网络分布所带来的问题,如消息通信、异构性、可伸缩性、资源共享等。在网络、操作系统之上的中间件系统为应用开发者和设计者提供了高层次的抽象接口,将许多复杂问题的处理细节隐藏起来。已有的中间件技术都遵循“黑盒”设计的思想,屏蔽了网络分布性、性能管理等的实现细节,使整个系统呈现为单一、整体的计算设施,保证了分布性在中间件之上的透明性。

随着 Internet 的大规模普及、移动设备和无线通信技术的快速发展,网络计算正从常规的有线网络环境向开放、移动的网络环境延伸,普适计算正逐步成为现实^[1]。与传统的分布式应用一样,新一代的分布应用也需要适合的中间件平台支持,以保证应用开发和运行的高效性。然而,现有的网络环境由于自身的特性,使得其计算模式与常规的分计算不同,限制了传统中间技术在开放、移动网络环境中的应用,进而对相应的研究提出了许多挑战。因此,本文对面向开放、移动网

络环境的新一代中间件技术进行了研究。

1 开放移动的网络环境

相对于传统的固定网络环境,现有的网络计算环境具有相当多的不同之处,这就导致了新一代中间件系统的研究必然面临更多的问题和挑战,主要来自开放性、移动性和便携式设备等方面。

开放性和移动性是现有网络计算环境的两大基本特征。开放性具有两方面的含义^[1]:一方面,Internet 连接允许分布式资源间的网络通信跨越网络边界;另一方面,无线通信允许用户持有的设备根据用户的移动模式,自由地加入和离开网络。移动性^[2]则意味着,无论是提供服务、资源的设备,或者需要访问资源的客户,位置可能都经常变化,整个网络的计算资源、连同网络本身都是动态变化的。

另外,从便携式设备方面看,移动中的设备经常会面对临时的、突发性的网络连接中断,甚至连支持网络通信的基础设

到稿日期:2009-04-10 返修日期:2009-07-02 本文受科技部十一五科技支撑项目——“现代服务业服务交互支撑平台”(项目编号:2006BAH02A0407)资助。

李琪林(1973-),男,博士,主要研究方向为可信计算、中间件;王敏毅(1973-),男,博士,主要研究方向为移动计算、智能 Agent;周明天(1939-),男,教授,主要研究方向为计算机网络、分布对象技术、并行分布处理。

施也可能突然改变,而且移动设备往往需要在一种非常不确定的网络环境中发现其它的主机或资源。同时,移动设备的资源往往比较有限,例如电池供电能力弱、CPU 速度不够、内存容量少,因此要求应用的计算开销不应很大;但矛盾的是,环境的多变性要求应用能快速响应频繁的变化,例如新的主机位置、新的通信方式、网络带宽的波动等。相对固定网络中运行在稳定主机上的应用而言,开放、移动的分布式应用充满着动态、不确定的因素^[2]。

2 新的技术需求

如前所述,开放和移动是今天网络环境的关键特征,相应地对中间件系统提出了新的技术需求,如图 1 所示。

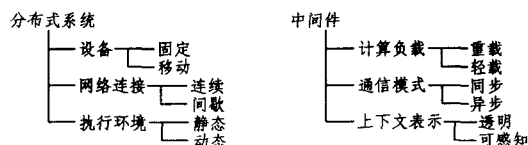


图 1 分布式系统和中间件的技术需求

从分布系统角度来看,设备、网络连接、执行环境是 3 个具有特征性的典型因素;如果针对中间件,则需要进一步考察计算负载、通信模式和上下文表示等特征^[3]。

在常规网络环境中,主机往往是固定的,通过高速、稳定的线路与外界连接,且系统运行在静态环境中。分布式应用的设计经常需要考虑以下的非功能性需求:伸缩性、开放性、异构性、容错能力和资源共享等。相应地,中间件通常需要承受比较重的计算负载,分布式协作宜采用同步方式,网络通信、应用管理、负载平衡等复杂处理机制往往对应用透明^[2,3]。

在开放、移动的网络环境中,分布式系统及相应的中间件在设计上则应有不同的考虑。首先,在各种资源限制下,设备上运行的中间件应是轻量级的,不应占用过多资源;其次,由于网络连接的不稳定性,异步协作方式往往更加有效,因此中间件及上层应用的设计可以多考虑消息队列、基于内容的协作或对等协作模型,而减少同步调用等待;第三,开放、移动的分布系统大都在动态的环境中执行,中间件一般不能根据网络的高度动态变化而对应用做出透明的调整并同时保证最佳的服务质量,因此比较适当的方式是将中间件与应用相结合,共同完成对环境的自适应调整^[2,3]。

3 新一代中间件面临的主要挑战

3.1 上下文感知

上下文感知主要解决的问题是应用发现和利用环境信息^[5]。这方面的研究并不是新近出现的,但仍存在相当多的工作需要开展。狭义上下文感知的研究构成了其它研究工作的基础,如借助上下文感知的自适应;广义上下文感知研究则直接用于许多移动应用,如车载电子地图系统。

上下文是应用所关注的环境状态和设置的集合,决定应用的行为或应用事件发生的场合^[5]。作为应用运行环境中任何可以影响应用行为的因素,上下文大体分为以下几类:1) 用户上下文,如用户映像、位置、邻近用户,甚至社会地位等;2) 设备上下文,任何与应用运行的物理设备相关的因素,如内存、电量、屏幕大小、输入设备、处理速度等;3) 环境上下文,除

物理设备以外的因素,如网络接口、带宽、协议、当前位置等。

上下文感知一般可从以下两方面定义^[2,5]:主动上下文感知,应用自动根据已发现的上下文信息改变自身行为;被动上下文感知,应用将新的或刷新的上下文提交给感兴趣的用户,或使这些信息可被用户检索。其中,主动方式在开放、移动的网络环境中有更好的应用,更适合今天的网络计算环境。

从应用的角度,上下文感知可分为以下几种类型^[2,5]:就近选择,附近的对象被突出和优先选择;自动上下文重配置,即根据环境变化修改系统的组件配置;上下文信息和命令;上下文触发的行为,类似于 IF-THEN 规则规定系统在不同环境下的动作。也有一种方法用于对上下文感知的关键特征进行分类,包括感应、调整、资源发现、数字化等,并将其归结为几种操作,即信息和服务的表示、服务的自动执行、上下文信息标记等。

对这类系统而言,感知上下文是非常重要的基础技术^[2,5]。最重要、使用最多的感知元素是位置,按照室外和室内可以有不同的方法。和位置同处于低级别的其它上下文信息包括时间、附近对象、网络带宽等。而更高级别的信息包括用户的当前活动、社会环境,这类信息往往比较难于获取。另外,上下文的变化本身也需要被感知。

为了支持上下文感知,中间件系统将应用和上下文感知子系统进行了必要分离,其功能是集成不同的感知技术,收集原始的传感器信息,并将其转换为应用可理解的格式,提供给上层公共的接口,并分发到感兴趣的应用。

上下文感知的主要难点在于对上下文信息进行表达和建模^[2,5]。目前的多数系统并没有统一的模型用来表达上下文信息,彼此的信息交换也比较困难,而且多数研究仅局限于位置信息。典型的表达模型包括符号化模型和几何模型,相应的数据结构主要是键-值对、标记编码、对象等。

3.2 环境自适应

考虑到开放、移动网络环境具有高度的动态性和有限带宽的特点,因此新一代的中间件必须解决动态变化的问题,即根据它们执行的环境自适应地调整自身的行为。

自适应的策略通常具有两个极端:对应用透明的自适应和完全依赖应用实现的自适应,如图 2 所示^[6]。对应用完全透明的自适应策略可以实现对现有应用的良好兼容,因此不需要专门调整就可运行在移动网络环境中。它具有应用开发简单,不需过多考虑移动环境特殊性的优点。但这类完全透明的系统级自适应方法存在固有缺陷,在某些场合下自适应效果并不理想,甚至会产生副作用,因此这种方法往往以牺牲功能或性能为代价。完全通过应用实现的自适应方法则是由应用负责响应移动资源的变化^[2]。

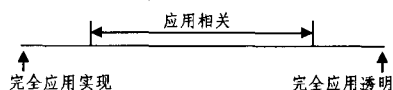


图 2 自适应策略的范围

上述两个极端策略都有局限性,理想的方法是进行折衷,即中间件系统保证主要的自适应优化,对于必须应用参与的调整,则通过与应用相关的特殊接口进行。例如通过系统监视资源水平,将变化通知应用,迫使其做出资源分配决定。

衡量系统自适应能力的指标之一是灵敏性(agility)^[7],即应用对环境变化的响应能力。由于可变因素比较多,因此

该指标是复合性的。但灵敏性往往和另外一个指标即稳定性(stability)矛盾,过度灵敏的系统往往容易导致系统不稳定。而且,频繁监测环境的变化以求良好的灵敏性,会带来较大的计算开销。这方面,控制理论可以发挥一定的作用。

从设计角度,一种提高系统自适应调整能力的通用方法是反省(reflection)^[3]。这种技术早期主要应用在编程语言和操作系统中。近年来,许多研究者开始在分布式系统及中间件中对反省技术进行研究,以期望提高这类系统的动态定制和自适应能力。反省的作用在于为中间件平台引入更多的开放性和灵活性。其本质是通过某种手段将系统内部原本对应用透明的一些数据、结构甚至行为揭示出来,使外部可以访问甚至修改。这样的系统往往具有较高的灵活性,可以根据环境的变化进行动态调整^[2]。

3.3 开放式协作

移动通信存在两个重要的特点:间歇性的非连接状态、通信带宽相对较低。这就决定了开放移动网络环境中的分布式交互应当是松耦合的,这与常规系统有着显著的不同。

从应用角度考察两种典型的分布交互模式:消息传递、客户-服务器,如图3所示^[2]。不难看出,前者的分布应用是消息驱动的,不存在显式的交互过程,因此属于松散耦合,适合于开放、移动的网络环境;而后的实质在于客户访问服务器的数据或服务资源,如果不做特殊处理,在非连接状态将使数据或服务资源对客户不可用,在弱连接的网络中将使交互开销过大。

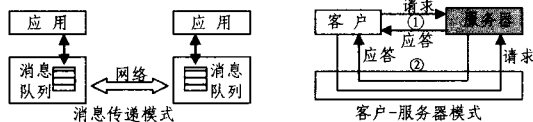


图3 两种分布交互模式的比较

针对分布交互模式问题,中间件系统大致可以采用3种方法^[1]:一是在核心使用消息传递的通信机制,但保持应用交互语义不变;二是 tuple 空间;三是引入更加灵活的协作模型,如基于内容的协作或对等(Peer to Peer)协作等。

Tuple^[2,8]空间技术早期出现于一种用于并行编程的协调语言系统 Linda 中,但近期研究发现它在移动通信中也有应用价值。tuple 是一种存储类型化数值向量的数据结构,维护若干 tuple 的数据空间可称为 tuple 空间。tuple 被创建后,通过 write 原语写入空间,以后可以通过 read 和 take 原语进行并行的访问。tuple 空间可以看作全局共享数据空间,它的存在解除了分布交互方的耦合。tuple 空间的主要缺点在于不支持上下文感知,且同步能力相对较弱。

针对分布式交互,另外一种典型方法是基于内容的协作或对等协作。内容协作^[1]类似于电子邮件系统,采用发布-订阅模式,实现分布式交互。在内容协作模型中,发送者根据订阅者的需要而非具体的网络地址,将类型吻合的消息发送给后者,无需集中协调者的介入,类似于报刊的订阅和分发。

与内容协作不同,对等协作^[1]允许参与者互相利用彼此的资源,完成计算任务,无需经过中间实体。在对等模型中,参与者既是服务和内容的提供者,又是服务和内容的消费者。相对于传统的客户-服务器模式,对等协作模型中的参与者彼此可以直接通信,每一个参与者既是客户端又是服务器。这种通信无需依赖网络中的某个集中式服务器或资源就可完

成。由于参与者间彼此连接,即使某一参与者出现问题,其余参与者仍旧可以继续通信,完全不受影响。

显然,内容协作和对等协作支持无中心的分布式交互模式,更适合于开放、移动的网络计算环境。相应地,新一代的中间件系统能支持这种开放式的协作模式,以提供更大的灵活性。

3.4 服务发现

服务发现也是开放、移动网络环境中的一个重要问题,由于主机的移动和链路的多变性,服务不再固定而稳定,因此新一代的中间件系统必须支持快速的服务动态发现^[2]。

具体而言,需要解决的问题包括服务如何动态发布、服务信息的失效、何处可以查询附近的服务等^[2]。大体上,移动服务管理可以采用两种方式^[2]:集中方式,在全局目录服务中维护动态的服务信息;分布方式,可以在各个移动区域的相对固定点跟踪附近的主机服务。后者在处理上许多地方类似 Mobile IP。

服务的发现机制可以是主动发现和被动发现,根本的区别在于提供者与发现者之间的交互关系^[2]。在主动发现过程中,发现者周期性地发出广播消息,在一定网络范围内的资源提供者一旦收到广播消息,会发送回应,将自己的资源信息告知对方。与之相反,在被动发现过程中,资源提供者处于主动方,负责将自身提供的服务资源通知发现者。

3.5 普适互操作

随着不同类型中间件系统的出现,中间件系统所采用的通信协议的多样性导致了异构中间件系统的互操作性问题。在开放、移动的网络计算环境中,系统不可能事先预知用于访问分布资源的通信协议,因此普适互操作性也就成为新一代中间件必须着力解决的关键问题^[1]。

对于普适的互操作问题,一种可能的方法^[1]是中间件网关。中间件网关通过在不同中间件系统之间解释相关的通信协议来实现互操作。在 ESBs (Enterprise Service Buses) 系统^[1]中,开发出一种更加灵活的基于中间件网关的互操作性方法,其中引入的关键概念是 ESB。ESB 是一种服务器基础设施,集成了一组可复用的网关。作为中间件系统间的一种中介,它负责解释不同系统支持的通信协议,实现异构中间件系统间的互操作。另一种典型的方法是反省^[1]。借助反省技术,中间件系统可以根据执行上下文,动态地插入(Plugging-in)最适合的通信协议,实现中间层的互操作。

结束语 今天的网络计算环境具有开放性和移动性的特点,常规的中间件技术不再适合,许多问题对其构成了挑战,需要研究新一代的中间件技术。本文首先揭示了现有网络环境的关键特征,在此基础上分析了新一代中间件系统需要满足的技术需求。随后,从所针对的问题角度对新一代中间件系统面临的主要挑战进行了详细的讨论,包括上下文感知、环境自适应、开放式协作、服务发现和普适互操作等,给出了一些解决方案。我们相信对这些问题的解决将直接推动未来中间件技术的发展。

参考文献

- [1] Issarny V, Caporuscio M, Georgantas N. A Perspective on the Future of Middleware-based Software Engineering[M]// Briand L, Wolf A, eds. Future of Software Engineering 2007. IEEE-CS Press, 2007

(下转第 132 页)

Chen信任模型所覆盖。这主要是因为它引入了不确定性的原因。它不会因为不知道而拒绝,因此给大多数诚实的实体提供了更多的合作机会,从而提高了网络的合作成功率。

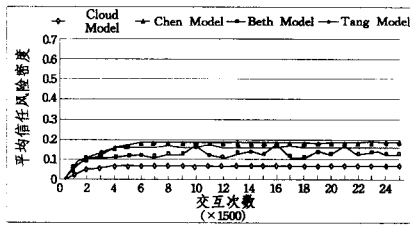


图1 平均信任风险密度实验结果

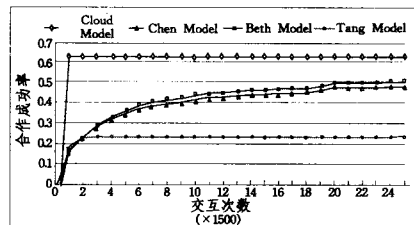


图2 合作成功率实验分析

结束语 本文从风险是由动态变化产生的这一角度,定义了无线传感器网络环境中的“风险信号”,进而定义了什么是“风险”。为降低无线传感器网络环境中的伪判定率,提出了新的风险信号评估方法,并改进了云信任模型。在对风险信号的评估中,鉴于上下文信息异常出现的不确定性,借用云模型理论来判定风险信号的发生。

在用多种不同上下文信息的动态变化来共同决定“风险”的发生这一问题上,本文提出的乘权重相加的方法,实际上没有考虑不同上下文信息之间动态变化的关联性。如果采用多维云模型来描述上下文的动态变化,将每个上下文信息作为云滴的一维,将更能反映问题的本质;同时在判断信任风险的过程中如何结合信誉的问题,这些都将是我们在未来一段时间所要做的工作。

参考文献

[1] 何锐,牛建伟,胡建平. 一种开放网络环境中的不确定信任模型[J]. 北京航空航天大学学报,2004,30(11):1125-1128
 [2] 李德毅,杜鹃. 不确定性人工智能[M]. 北京:国防工业出版社,2005
 [3] Blaze M, Feigenbaum J, Lacy J. Decentralized trust management [C]//the Proceedings of the IEEE Symposium on Security and Privacy. USA: IEEE Computer Society Press, 1996: 164-173

[4] Josang A. The right type of trust for distributed systems[C]// Proceedings of the 1996 New Security Paradigms Workshop. California, USA, 1996: 119-132
 [5] Beth T, Borcherding M, Klein B. Valuation of trust in open networks [C] // Proc. of the Conf. on Computer Security. New York: Springer-Verlag, 1994: 3-18
 [6] 唐文,陈钟. 基于模糊集合理论的主观信任管理模型研究[J]. 软件学报,2003,14(8):1401-1408
 [7] Chen Haiguang, Wu Huafeng, Zhou Xi, et al. Reputation-based Trust in Wireless Sensor Networks[C]//Proc. of International Conference on Multimedia and Ubiquitous Engineering (MUE'07). 2007: 603-607
 [8] 田春歧,邹仕洪,王文东,等. 一种新的基于改进型 D-S 证据理论的 P2P 信任模型[J]. 电子与信息学报,2008,30(6):1480-1484
 [9] Jameel H, Hung L X, Kalim U, et al. A trust model for ubiquitous systems based on vectors of trust values[C]//Proc. of the 7th IEEE Int'l Symp. on Multimedia. Washington: IEEE Computer Society Press, 2005: 674-679
 [10] Han Guangjie, Choi D, Lim W. A Reliable Approach of Establishing Trust for Wireless Sensor Networks[C]//Proc. 2007 IFIP International Conference on Network and Parallel Computing. 2007: 232-237
 [11] Sun Yan, Han Zhu, Liu K J R. Defense of Trust Management Vulnerabilities in Distributed Networks[J]. IEEE Communications Magazine, Feature Topic on Security in Mobile Ad Hoc and Sensor Networks, 2008, 46(2): 112-119
 [12] Melaye D, Demazeau Y. Bayesian dynamic trust model [C] // LNCS 3690. Berlin: Springer-Verlag, 2005: 480-489
 [13] Kyung T, Seo H S. A trust Model using Fuzzy Logic in Wireless Sensor Network[J]. Proc. of World Academy of Science, Engineering and Technology, 2008, 32(8): 69-72
 [14] Liu Ke, Nael A-G, Kang K-D. Location verification and trust management for resilient geographic routing[J]. Journal of Parallel and Distributed Computing, 2007, 67(3): 215-228
 [15] Manchala D W. Trust Metrics, Models and protocols for electronic commerce transactions [C] // Proc. of the 18th International Conference on Distributed Computing Systems. Washington, DC, USA: IEEE Computer Society, 1998: 312-321
 [16] Josang A, Presti S. Analysing the relationship between risk and trust [C] // Proc. of the iTrust '04. LNCS 2995. Oxford, UK, 2004: 135-145
 [17] 张润莲,武小年,周胜源,等. 一种基于实体行为风险评估的信任模型[J]. 计算机学报,2009,32(4): 688-698

(上接第 78 页)

[2] 王敏毅. 面向移动计算环境的分布对象技术[D]. 成都:电子科技大学,2002
 [3] Capra L, Emmerich W, Mascolo C. Middleware for Mobile Computing[R]. UCL Research Note RN/30/01. Submitted for publication, July 2001
 [4] Baggio A. Adaptable and Mobile-aware Distributed Objects [M]. Université Pierre & Marie Curie, June 1999
 [5] Chen Guanling, Kotz D. A Survey of Context-aware Mobile Computing Research [R]. Dartmouth Computer Science TR20

00-381. 2000

[6] Satyanarayanan M. Fundamental Challenges in Mobile Computing [C] // Proceedings of ACM PODC '96. 1996
 [7] Noble BD, Satyanarayanan M, et al. Agile application-aware adaptation for mobility [J]. ACM SIGOPS Operating System Review, 1997, 31(5): 276-287
 [8] Davis N, Friday A, Wade S, et al. A distributed systems platform for mobile computing [J]. Mobile Network Application, 1998, 3(2): 143-156