

支持广域网的协作式服务发现机制^{*})

张发恩 金蓓弘 齐凤亮

(中国科学院软件研究所软件工程技术中心 北京 100080)

摘要 服务发现是网络分布式环境下进行信息共享、数据集成、流程协作的前提。广域网下的服务发现机制必须在无可直接利用的广播和组播机制上解决系统的规模伸缩性问题,以合理的代价为用户提供高效的服务发现机制。采用覆盖网络体系结构,并实际构建了具有一定可伸缩性的服务发现系统 Service CatalogNet。Service CatalogNet 基于分布存储的服务信息和后缀树形式的服务信息索引,实现了协作式服务发现机制,特别提供了 QoS 感知的服务发现机制,即可基于客户的 QoS 指标生成应用层选播/多播路由,进而实现服务发现。实验结果表明,该服务发现机制在发现性能上优于 LDAP 实现。

关键词 服务发现,索引后缀树,QoS 感知的协作发现

Collaborative Service Discovery Mechanism for Wide Area Network Environment

ZHANG Fa-en JIN Bei-hong QI Feng-liang

(Technology Center of Software Engineering, Institute of Software, Chinese Academy of Sciences, Beijing 100080, China)

Abstract Service discovery is the premise of information sharing, data integration and process collaboration in distributed environment. Under the wide area network environment in which no suitable broadcast and multicast mechanism is available to service discovery mechanism, achieving scalability with respect to size is required, in addition to providing efficient discovery performance while keeping appropriate costs. This paper adopts an overlay network architecture and constructs a service discovery system, named Service CatalogNet, aiming at a certain degree of scalability. Based on distributed service information and index of suffix tree, Service CatalogNet implements collaborative service discovery mechanism, especially supporting QoS-aware discovery, i. e. generating anycast/multicast routing path first according to client's QoS metrics, and then complete collaborative discovery. Experiment results show the discovery performance of our service discovery mechanism is more efficient than that of LDAP implementations.

Keywords Service discovery, Index suffix tree, QoS-aware collaborative discovery

1 引言

服务发现是网络分布式环境下进行信息共享、数据集成、流程协作的前提。网络用户可以在互连的网络环境中发布可共享的服务信息,或者查询自己需要的服务资源,以便进行进一步的信息处理。

目前学术界和工业界在服务发现方面进行了不少的探索和研究。局域网环境下的服务发现方案已日臻完善并且应用到了实际系统之中。但是在广域网环境下必须面对下列具有挑战性的问题:第一,与局域网环境不同,没有可被服务发现机制直接使用的广播和组播机制;第二,广域网内存在大量的设备和服务,服务发现必须能适应服务数量和并发用户的变化,具有对服务规模和并发用户请求数量的可伸缩性;第三,更强调以合理的代价,包括通讯开销、计算开销和存储开销(含服务复制开销),去获得用户可接受的服务发现响应时间;第四,用户需求更呈现多样性,例如移动用户对其提交的服务发现请求还附带一些 QoS 指标,以限制服务发现的过程和结果。

目前,广域网服务发现的基本途径是在集中型发现模式

的基础上扩展形成结构化分布式发现模式,建立起各服务目录服务器间的服务信息交换或查询请求转发机制,进而实现广域网环境下有效的服务发现。已有的面向广域网环境的服务发现协议有:SSDS^[1], INS/Twine^[2], Superstring^[3], CSP^[4], GloServ^[5]等。加州大学 Berkeley 分校为分布式平台研究项目 Ninja 开发了一个安全的服务发现服务 SSDS(Secure Service Discovery Service),为广域网中定位服务提供了一个高可用的、容错的、持续可扩展的服务。在网络可伸缩性支持方面,它构建了一个自适应的多层树状服务体系架构,在此基础上,SSDS 还采取了描述聚合与查询路由策略。INS/Twine 是麻省理工学院构建的一个可扩展的服务资源发现系统,用于解决大规模网络中的服务资源发现问题。它的主要特征有三个:一是服务描述的表达能力强,能够处理复杂的资源描述和查询模式;二是能够处理操作环境的动态性,包括资源状态和网络节点的变化;三是可伸缩性好,能够达到包括大量资源、跨广域网的规模。Superstring 是一个适用于广域网环境的 P2P 服务发现协议,它使用分布式哈希表 DHT 来构造 P2P 方式的覆盖网络,因此也能够很好地支持网络可伸缩性。但 QoS 感知的服务发现机制,在上述几个著名的协议中

^{*})本文研究得到国家高技术研究发展计划 863 资助项目(编号 2006AA01Z231)和国家自然科学基金(编号 60673123)的资助。张发恩 硕士研究生,研究方向为分布式计算、软件工程;金蓓弘 博士,研究员,研究方向为分布式计算、软件工程;齐凤亮 硕士研究生,研究方向为分布式计算、软件工程。

均没有被提及。

本文以分布式系统中服务资源的发现机制为出发点,研究广域网环境的特点给服务发现带来的影响,构建具备一定可伸缩性、支持 QoS 感知的服务发现系统 Service Catalog-Net。

2 服务信息的分布存储

Service CatalogNet 采用覆盖网络体系结构,如图 1 所示。系统中的服务器构成了应用层覆盖网络的结点,这些结点之间是对等的关系,它们基于 HTTP 协议和语义内容进行通信。覆盖网络中,目录服务器(DS)用来存储共享的服务信息,对外提供服务信息的注册、更新、删除与查询等功能。而目录管理服务器(DSA,包括主 DSA 和从 DSA)负责管理系统内所有的 DS。同时管理系统的用户(包括服务提供者和服务请求者)信息、管理系统内所有服务的索引信息、维护系统覆盖网络的状态信息,等等。

这样,覆盖网络作为一个基础平台向服务请求者和服务提供者提供服务。服务请求者根据所需服务的属性在覆盖网络中声明服务请求,服务提供者注册服务,服务信息存储在覆盖网络中的多个目录服务器上,后者响应客户的服务请求。

为了支持系统的伸缩性,我们把服务信息分布存储在覆盖网络的 DS 中,同时把服务信息的索引保存在 DSA 的后缀树(Suffix Tree)^[6]形式的索引 SuffixIndex 中。

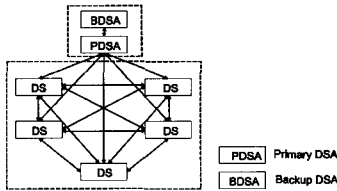


图 1 目录服务器组成的覆盖网络

我们定义字符集 Σ 上的串 S 对应的后缀树 $ST = \{ root, V, E, T, \Sigma \}$, 其中 $root$ 表示树的根节点, V 表示树中所有的节点且 $root \in V$, E 表示树中标记边的集合, T 表示树中叶节点集合, Σ 为字符集合。且 $T \subseteq V, E \subseteq V \times \Sigma \times V$ 。 $\forall n \in V$, 定义 n 的路径为 $l_1 l_2 \dots l_k$, $l_1 l_2 \dots l_k$ 是起始于 $root$ 终止于 n 的路径上所有边上字符的串联, ST 中所有节点都有唯一的路径。由于 Σ 上的字符串 S 的任意一个后缀都可以在对应的 ST 中找到一个节点与其对应, 故称 ST 为 S 的后缀树 (Suffix Tree)。

鉴于我们使用 XML 语言的一个子集描述服务模型^[7], 所以使用上述后缀树来建立 XML 文档的索引结构, 即把出现在 XML 文档中的路径看成字符串为其建立索引。

构造 XML 文档索引后缀树的基本思想是: 首先将 XML 文档表示成树形, 用符号 XT 表示; 然后, 对该树构造一棵包含每个叶节点路径串的所有后缀串之聚合后缀树, 称该后缀树为对应 XML 文档的索引后缀树 SuffixIndex。由于树的所有内部节点的路径是叶节点路径的子串, 因此只需要对叶节点路径构造后缀树即可。下面给出 SuffixIndex 的定义。

设 $\sigma(XT) = \{ S_1, S_2, \dots, S_L \}$ 为 XT 中所有叶节点的路径集合, 那么 SuffixIndex 是为所有 $S_i (i = 1, 2, \dots, L)$ 构造形成的后缀树 ST 。图 2 给出了一个服务描述的 XML 文档片断以及对应的索引后缀树示例。

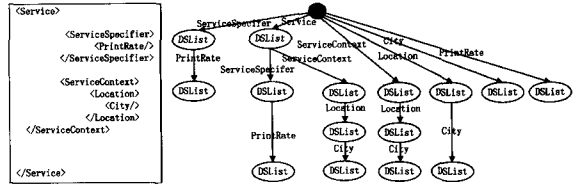


图 2 一个服务描述的 XML 文档片断以及对应的索引后缀树

索引后缀树 SuffixIndex 中的 DSLIST (即图 2 中节点内的数据) 表示服务与存储服务的 DS 之间的映射关系。

为了提高 SuffixIndex 的构造效率, 还引入了后缀链 (Suffix Link)。设 SuffixIndex 中节点 v 的路径为 $x\alpha$, 其中 x 为路径中的基本单元 (如路径 Service, ServiceSpecifier, Price 中的 Service, ServiceSpecifier, Price), α 为子路径串。若 SuffixIndex 中有节点 $s(v)$ 的路径为 α , 则有一指针自 v 指向 $s(v)$, 称该指针为节点 v 的后缀链。在 SuffixIndex 中, 所有节点均有后缀链指向该节点的后缀节点, 特别是 SuffixIndex 中根节点的所有子节点的后缀链均指向根节点。也就是说, 根节点表示所有串的空后缀, 根节点的后缀链为空。

考虑到服务描述信息和索引信息的上述分布存储方式, 系统是以事务方式执行服务信息的注册、更新和删除三类操作, 以保证服务信息的一致性。

3 服务的协作发现机制

用户登录系统时, 系统会为其分配一个接入目录服务器 (Access Directory Server, ADS) 用于处理后续的请求。ADS 在收到用户的查询请求后, 负责发起和控制协作发现过程:

- (1) ADS 首先执行本地服务匹配算法, 如果在本地找到匹配的服务, 则直接返回给用户;
- (2) 如果用户查询中提供服务提供商的标识, 那么 ADS 调用 DSA 上的 ADS 选择算法获得与该服务提供商绑定的 ADS, 然后 ADS 将该查询请求发送到这个 ADS, 由这个 ADS 执行本地服务查找, 该 ADS 会将查找结果返回给用户;
- (3) 否则, 进入 QoS 感知的选播/多播式的服务发现过程:

(3.1) 选择候选 DS 集合。DSA 根据用户的查询条件和 SuffixIndex 信息选择出将要处理查询的候选 DS 集合。

(3.2) 生成 QoS 感知的选播/多播式查询路由。ADS 根据服务请求者选择的查询方式 (应用层选播服务查询或应用层多播服务查询)、请求者要求的 QoS 指标和它所维护的系统网络状态信息, 对 DSA 返回的候选 DS 集合生成对应查询方式的路由路径或多播树, 之后查询条件将会在得到的路由路径或多播树上传播。

(3.3) 进行本地服务匹配。当服务查询请求到达路由路径或多播树上的 DS 时, 需要运行本地服务匹配算法, 本地服务匹配算法将会对每个查询结果给出一个匹配度。

(3.4) 服务选择。在选播方式下, ADS 会选择用户要求个数的或者合适个数的查询结果。在多播方式下, DS 会首先等待多播树上它的所有子节点的查询结果, 并聚合本地发现的服务, 之后根据服务的匹配度和服务本身的 QoS 信息对以上所有查询结果进行进一步的选择, 然后将用户要求个数的或者合适个数的服务向上返回给它的父 DS 节点。重复这个过程直到到达初始 DS (即多播树的根节点, 也即 ADS), 最后由 ADS 将结果返回给用户。

上述过程涉及到多个算法,如根据查询条件从 DSA 获取候选 DS 集合的 Service_Query_From_SuffixIndex 算法、应用层选播式服务查询路由生成算法 QoS-Aware_Anycast_Query_Routing、应用层多播式服务查询多播树生成算法 QoS-Aware_Multicast_Query_Routing。图 3 给出了 QoS-Aware_Multicast_Query_Routing 算法的框架。

Input: Message, CandidateDSSet, QoS Metrics

Output: Service list

Procedure QoS-Aware_Multicast_Query_Routing

1. MulticastTree mt ← Gen_ConstrainedLoadShortedPath_MulticastTree(QoSMetrics, CandidateDSSet) or Get MulticastTree From Father Node
2. List(Service)queryResults;
3. Set(MulticastTreeNode)childrenSet ← the current DS's children nodes set on mt
4. Foreach MulticastTreeNode s in childrenSet
5. Send the query message and MulticastTree mt to s, get the query results from s, and add to queryResults
6. End of Foreach
7. return queryResults
8. End of Procedure

Procedure Gen_Constrained_Load_Shortest_Path_MulticastTree (QoSMetrics, CandidateDSSet)

1. Set (DS) cDSSet ← Scan CandidateDSSet with Single-DS-Constrained QoS Metrics, and delete the unsatisfied DS from CandidateDSSet, delete these Single-DS-Constrained QoS Metrics
2. Set (DS) resultDSSet ← Null
3. While (cDSSet ≠ Null)
4. Get the DS s in cDSSet which has the best value about the remained QoS Metrics, add s to resultDSSet and delete s from cDSSet
5. Foreach DS s' in cDSSet
6. If s' has better value about the remained QoS Metrics, update it
7. End of Foreach
8. End of While
9. MulticastTree mt ← Generate an tree from resultDSSet
10. Return mt
11. End of Procedure

图 3 多播式服务查询路由算法描述

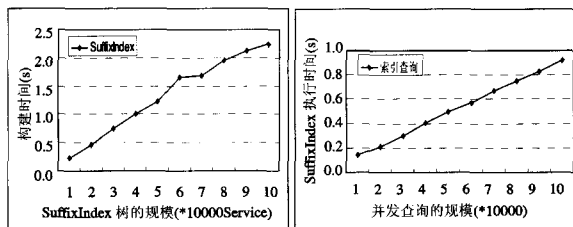
4 实验与比较

我们分别对索引后缀树 SuffixIndex 的性能和开销、协作服务查询机制的性能进行了实验评估。

首先,测试构建 SuffixIndex 所需要消耗的时间与内存情况。实验中,服务描述信息根据模板随机生成,实验结果如图 4(a)所示。由图 4(a)可以看出,在构建 SuffixIndex 的过程中,构建时间随构建的服务数量近似线性增长,构建时间稳定在 0.238s/10000Service 左右,消耗的内存 ≤ 2MB,符合我们的预测。

接着,测试利用索引进行查询操作的性能。对上一步构建的规模为 100000Services 的 SuffixIndex 结构,并发执行一定数量的查询操作,统计所需要消耗的时间。实验中,查询内容按照我们规定的查询条件的 BNF 范式随机生成,实验结果如图 4(b)所示。从图中可以看到,对 SuffixIndex 中的服务索

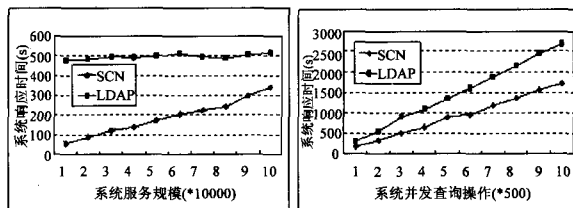
引查询时间随并发查询操作数量增加而呈线性增长。在服务规模为 100000 的 SuffixIndex 上,平均每 10000 个并发查询操作,所需时间仅为 98.2ms,可见 SuffixIndex 的查询性能是比较高效的。



(a) 构建 SuffixIndex 数据结构的性能 (b) 在 SuffixIndex 数据结构上执行查询操作的性能

图 4 SuffixIndex 的性能和开销测试

最后,我们测试了协作服务查询机制的性能。LDAP^[8]是在 X.500 协议简化的基础上演变而来的目录服务协议,也可用于广域网的服务发现。我们选择了 IBM Tivoli v5.2(主从模式)作为 Service CatalogNet 的对比软件。为了达到比较的等价性,两套系统各部署在 3 台服务器上,这 6 台机器在 CPU、内存、操作系统等配置上完全一致。考虑到 LDAP 不支持语义匹配,测试实验比较了常规服务发现请求的系统响应时间,比较结果如图 5 所示。



(a) 固定系统服务的规模:10000Service/ 每台测试机 (b) 固定并发查询操作数量:1000 次

图 5 Service CatalogNet(在图中表示为 SCN)与 LDAP 执行协作服务查询操作的性能比较

其中(a)表示系统响应时间随并发查询操作数量增加的变化,(b)表示系统响应时间随系统规模增长的变化。图 5(a)所示的是两系统响应时间随并发操作数量的变化,即在服务规模为 10000 个服务/每台测试机的基础上,由 1 台测试机发起进行查询,系统并发执行 500~5000 个查询匹配操作,我们的协作服务发现机制与 LDAP 查询两种方式各自所需要的时间。由图中可见,随并发操作数量的增加,两种方式的操作响应时间均线性增加。Service CatalogNet 的执行性能平均为 342.217s/1000 次服务查询,优于 LDAP 查询的平均性能 524.308s/1000 次服务查询。

图 5(b)所示的是两系统响应时间随系统规模增长的变化,即 3 台测试机并发执行 1000 个服务查询,系统服务规模从 1000 个服务/每台测试机增长到 10000 个服务/每台测试机时两种服务查询并发操作时间的变化。由图中可见,随系统服务规模的增长,Service CatalogNet 的操作响应时间线性增加,以 54.816s/1000Services 为起点,平均每增加 1000 个服务/每台测试机,并发查询操作时间增加 32.058s;而 LDAP 的服务查询响应时间基本稳定在 506.54s/1000 次服务查询,与系统服务规模无关。上述两个实验说明,Service CatalogNet 面对变化的并发用户数量和服务数量具有一定的可伸缩性。

另外,假定 Service CatalogNet 需要并发进行 Q 个服务查询,且均匀选择 DS 来发起每一个协作查询,那么每个 DS 应该处理的查询百分比为 $1/ns * k$,这里, ns 表示系统中 DS 的个数, k 表示每个服务会分布存储到 k 个 DS 上。若 $ns = 25$, $k = 2$,则理论分析可得每个 DS 处理 8% 的查询。上述的实验结果也表明:80% 的 DS 处理的查询数低于总数的 10%,并且没有 DS 会超出平均值太多,所以查询在实际执行中可达到平均分布。

结束语 服务发现是网络分布式环境下进行信息共享、数据集成、流程协作的前提,针对广域网的服务发现还存在较多问题有待解决。本文主要对广域网环境下的服务发现机制进行研究,构建了一个具有一定可伸缩性、支持 QoS 感知的多播/选播式查询的服务发现系统 Service CatalogNet。我们对其中的关键算法进行了实验和比较。结果表明,基于服务信息的后缀索引结构,Service CatalogNet 在服务的协作查询性能上优于 LDAP 实现。

参 考 文 献

[1] Hodes T D, Czerwinski S E, Zhao B Y, et al. An architecture for secure wide-area service discovery. *Wireless Networks*, 2002, 8 (2/3): 213-230

[2] Balazinska M, Balakrishnan H, Karger D. INS/Twine: A scalable peer-to-peer architecture for intentional resource discovery//

International Conference on Pervasive Computing 2002. August 2002

[3] Robinson R, Indulska J. Superstring: A Scalable Service Discovery Protocol for the Wide Area Pervasive Environment // Proc. of the 11th IEEE International Conference on Networks. Sydney, September 2003

[4] Lee C, Helal S. A Multi-Tier Ubiquitous Service Discovery Protocol for Mobile Clients // Proceedings of the 2003 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2003). Montr' eal, Canada, 2003

[5] Arabshian K, Schulzrinne H. GloServ: Global service discovery architecture // *MobiQuitous*. IEEE Computer Society, June 2004: 319-325

[6] Grossi R, Italiano G F. Suffix trees and their applications in string algorithms // Proc. 1st South American Workshop on String Processing, 1993: 57-76

[7] Jin Beihong, Zhang Liang, Zang Zhi. A Unified Service Discovery Framework // the 6th International Conference on Grid and Cooperative Computing. Urumchi, Xinjiang, China, August 2007: 16-18

[8] Wahl M, Howes T, Kille S. Lightweight Directory Access Protocol (v3) // RFC 2251. December 1997

(上接第 237 页)

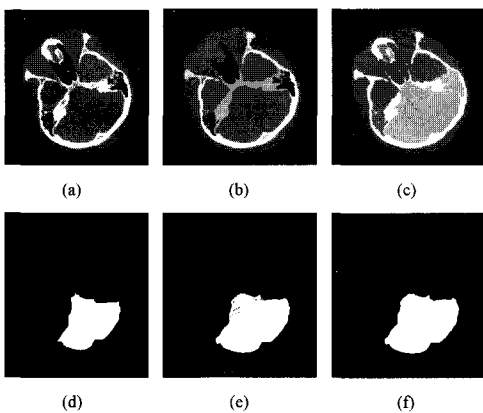


图 3 改进的水平集分割算法实验结果

图 3 中, a 为输入的原数据像; b 为通过噪音滤波处理后的图像; c 为通过阈值滤波器的数据图像; d 为传统 level set 分割的图像结果; e 为通过噪音滤波处理后进行分割的图像结果; f 为利用改进后的 level set 算法进行分割后得到的数据图像。选择合适的种子点及其它参数值,可以分别把目标图像从原图像中分割出来。

实验过程中,先用滤波器对图像进行平滑,以增强图像边缘并滤去噪声,然后根据区域之间的统计特性的相似度重新定义了 Level Set 方法的演化停止准则。我们分别应用改进前后的水平集方法对医学图像进行了分割,并使用了相同的初始化参数,得到的结果如图 3 所示。经过大量的实验,改进的算法无论是对于医学图像数据,还是对普通图像数据,都具有比较好的效果,同时,算法时间也得到了改善。

结束语 改进后的方法取得了良好的分割效果,算法的运行时间也得到了改善,对于边界比较模糊和灰度值接近的

医学影像具有很好的分割结果,较之传统的水平集分割方法有很大提高。由于医学图像的复杂性和多样性,决定了医学图像分割的算法也只能在一定的范围内有效,而不能普遍适用。进一步的研究将集中在继续改进水平集方法的速度函数,提高分割的可靠性方面。

参 考 文 献

[1] Sethian J A. A fast marching level set method for monotonically advancing fronts [J] // Proceedings of the National Academy of Sciences. vol 93, 1996: 1591-1595

[2] Harders M, Szekely G. Enhancing human-computer interaction in medical segmentation [J] // Proceedings of the IEEE. 2007, 91(9): 0018-9219

[3] Sethian J A, Adalsteinsson D. An Overview of Level Set Methods for Etching, Deposition, and Lithography Development [J]. *IEEE Transactions on semiconductor manufacturing*, 1997, 10(1)

[4] Sethian J A. *Level Set Methods and Fast Marching Methods* [J]. Cambridge University Press, 1996

[5] Malladi R, Sethian J A, Vemuri B. Shape modeling with front propagation; a level set approach [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995, 17(2): 158-174

[6] 朱付平, 田捷, 林瑶, 等. 基于 Level Set 方法的医学图像分割 [J]. *软件学报*, 2002, 13(9): 1866-1872

[7] 田捷. *集成化医学影像算法平台的理论与实践* [M]. 北京: 清华大学出版社, 2004

[8] 梁晓云, 罗立民, 曾卫明. 基于 Level Set 方法的医学图像分割 [J]. *电路与系统学报*, 2003, 8(6): 78-81

[9] 包尚联, 谢耀钦, 周晓东, 等. 基于医学影像计算机辅助诊断的分割方法 [J]. *中国医学物理学杂志*, 2003, 20(2): 83-86

[10] 吴月娥, 周康源, 李传富, 等. 基于 level sets 的医学图像分割 [J]. *北京生物医学工程*, 2006, 25(3)