基于QoS和多级索引的Web服务发布订阅

何 倩 李 佳 胡启伟 强保华

(桂林电子科技大学云计算与复杂系统重点实验室 桂林 541004)

摘 要 发布订阅机制有利于实现对大规模Web服务的主动管理,提出了基于QoS的Web服务发布订阅模型和系统架构,设计了基于QoS和多级索引的Web服务匹配算法。Web服务的QoS属性和订阅的属性约束所形成相应的匹配关系构成模型的关键;将发布的Web服务及其QoS和服务订阅一起生成过滤矩阵,通过属性约束覆盖可以减少重复匹配;按QoS属性类型对发布的Web服务建立多级索引,生成属性到服务的映射,可以实现服务订阅的快速匹配。实验结果表明,该Web服务发布订阅系统比传统方法有较大提升,能够适应于大规模分布式Web服务管理。

关键词 发布/订阅,Web服务,多级索引,服务质量,矩形过滤

中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.4.021

QoS and Multilevel Index Based Web Service Publish/Subscribe

HE Qian LI Jia HU Qi-wei QIANG Bao-hua

(Key Laboratory of Cloud Computing and Complex System, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract The publish/subscribe (P/S) can help to manage big scale web services actively. In this paper, a QoS based web services P/S model and system architecture with a web services match algorithm based on QoS and multilevel index was proposed. The matching relationship between QoS of web services and the subscription constraint is a key of the P/S model. The published web services and the service subscription are used to generate a filter matrix to reduce redundant matches. The multilevel index is constructed according to the QoS attribute type on the published web services, and then the map from attribute to service is generated to subscribe services rapidly. The experiments show that the proposed web service P/S system is better than traditional methods and is suitable to the large scale distributed web service management.

Keywords Publish/subscribe, Web service, Multilevel index, QoS, Rectangle filter

1 引言

随着云计算和云服务的普及,越来越多的互联网系统应用云计算技术来提高系统性能和改善用户体验。云计算按层次可划分成:基础设施即服务(IaaS)、平台即服务(PaaS)和软件即服务(SaaS)[1]。Web Service因具有平台无关、松耦合和良兼容等优势,已经成为云计算对外提供的最重要的接口封装形式,特别是在SaaS层,如何对大规模的Web服务进行有效的管理成了一个亟待解决的问题[2,3]。

发布订阅系统作为一种异步通信机制,具有松耦合、匿名、多对多通信和可扩展的特点,利用发布订阅来提高Web Service管理的主动性,已经成为业界研究的热点[4-6]。IFTWS框架[4]实现了发布服务的应用和使用服务的应用之间的数据交换;文献[5,6]分别给出了基于WSN规范族的Web Service发布订阅系统和基于发布订阅的Web服务QoS信息分发模型。发布订阅机制的引入提高了Web服务管理系统的主动性,但是目前还缺乏一个公认的基于QoS的Web服务发布订阅模型和架构。

实现发布的Web服务与服务订阅之间的快速匹配是Web服务发布订阅系统的关键。传统的暴力法是一种方式,同时也可以借鉴其他的发布订阅系统的匹配算法。Aguilera[7]等提出了一种基于搜索树的算法,该算法订阅维护的成本较低,降低了维护成本;Campailla[8]等提出一种基于二叉判定图(Binary Decision Diagram, BDD)的算法,该算法同时支持"与"和"或"两种操作;文献[9]提出了一种多维索引和计数法相结合的快速匹配混合算法;Siena[10]采用订阅覆盖的方式有效地缩小了订阅匹配的搜索空间;另外还有一些基于XML的匹配算法,如XFilter[11]、YFilter[12]等。以上方法在处理Web服务发布订阅时都需要根据QoS的特征重新设计,同时,互联网上存在大量具有不同QoS的候选Web服务[13,14],在海量服务中快速匹配满足QoS要求的Web服务仍然是一个难点。

到稿日期:2015-03-20 返修日期:2015-07-10 本文受国家自然科学基金(61201250,61163057,61163058),广西自然科学基金(2012GXNSFBA053174),广西高校科研项目(YB2014140),高校重点实验室基金(14102)资助。

何 倩(1979-),男,博士,教授,CCF会员,主要研究方向为服务计算、分布式计算,E-mail:heqian@guet.edu.cn;李 佳(1990-),男,硕士生,主要研究方向为分布式计算;胡启伟(1990-),男,硕士生,主要研究方向为分布式计算;强保华(1971-),男,博士,教授,CCF会员,主要研究方向为服务计算、大数据分析。

本文提出基于 QoS 和多级索引的 Web 服务发布订阅机制，针对 Web 服务的 QoS 属性提出发布订阅模型，将服务组织成一种矩形结构，迅速过滤掉大量不能被订阅匹配的服务，引入类似 Siena 的订阅覆盖思想，在 QoS 属性值之间建立覆盖关系，构造出一个多级索引的数据结构来加快匹配速度。实验表明，本方法比传统方法在扩展性和匹配速度上有所提高，适应于大规模分布式 Web 服务管理。

2 基于 QoS 的 Web 服务发布订阅模型

2.1 QoS 模型

为了度量 Web 服务的 QoS，将 Web 服务的 QoS 属性分为积极属性、消极属性和中性属性 3 大类[14]。积极属性考虑 QoS 最大化，如可用性、吞吐量；消极属性考虑 QoS 的最小化，如服务价格、响应时间等；中性属性考虑 QoS 一致性，比如服务功能、服务名称、WSDL 地址等。文献[14,15]给出了 Web 服务 QoS 的一些属性。在面向服务选择的发布订阅系统中，可以将 QoS 模型定义为 $QoS=\{RT,A,T,S,R,P,L,SN\}$，$RT$ 代表响应时间，A 代表可用性，T 代表吞吐量，S 代表有效，R 代表可靠性，P 代表价格，L 代表潜伏性，SN 代表服务名称。这些属性反映了 Web 服务的各方面的功能，能够帮助用户选择适合自己的服务。同时该模型也是可以扩展的，可以随时添加新的 QoS。

2.2 QoS 属性和 QoS 属性约束

将 QoS 属性表示为一个三元组（*property*，*type*，*value*），*property* 表示 QoS 属性名，用字符串表示，从 QoS 模型中取值；*type* 表示属性类型，用一个整型数据来表示，其中积极属性表示为 1，消极属性表示为 −1，中性属性表示为 0；*value* 为 QoS 属性值，用一个 float 或者 String 类型的值来表示，如 $QoS=\{RT,-1,2\}$ 表示响应时间小于 2ms。

利用属性覆盖思想将 QoS 属性组织成覆盖关系。若 $QoS_1=\{p_1,t_1,v_1\}$ 被 $QoS_2=\{p_2,t_2,v_2\}$ 覆盖，则必须满足：$\{p_1=p_2\}\cap\{t_1=t_2\}\cap\{\Theta(v_1,v_2)\}$ 为真。覆盖规则 Θ 如表 1 所列，当 QoS_1 和 QoS_2 的属性名和属性类型都相等时，若 QoS 属性值满足表中的条件，则表示 QoS_1 被 QoS_2 覆盖，记为 $QoS_1\subseteq QoS_2$。例如 $QoS_1=\{RT,-1,120\}$ 能被 $QoS_2=\{RT,-1,98.5\}$ 覆盖。

表 1 覆盖规则 Θ

Type	$QoS_1\subseteq QoS_2$
1	$v_1\leqslant v_2$
0	$v_1=v_2$
−1	$v_1\geqslant v_2$

将 QoS 属性约束 QoSC 也用一个三元组来表示，$QoSC=\{attribute,operate,value\}$，*attribute* 表示该约束所针对的属性的名称，操作符 *operate* 是约束对应的操作（例如 =，>，< 等），*value* 代表具体的匹配阈值。比如 $QoSC=\{T,>,100\}$ 表示要求吞吐量大于 100M。

2.3 Web 服务和订阅模型

Web 服务 PService 由集合 $\{QoS_1,QoS_2,\cdots,QoS_n,Id,mark\_p\}$ 组成，其中 QoS_i 表示某个 QoS 属性；*mark* 为服务标识，用来表示服务的结构，由一串固定长度的字符串组成；*Id* 是服务的唯一标记，*Id* 相同的服务表示同一个服务，不同服务的 *Id* 都是不同的，具体的 *Id* 值由命名服务器指定。

服务订阅是指服务请求者发起的对所需要的 Web 服务的要求，这些要求可表示为多个 QoS 属性约束的集合，即 $SService=\{QoSC_1,QoSC_2,\cdots,QoSC_n,mark\_s\}$。其中 $QoSC_i$ 表示 QoS 属性约束；*mark_s* 表示属性约束标识，与服务模型中的标识类似，用来表示订阅的结构，由一串固定大小的字符串表示。

2.4 匹配概念

定义 1 设服务 PService 有 $QoS_i=\{p,t,v\}$，服务订阅 SService 中有 $QoSC_i=\{pc,tc,vc\}$，若 $\{p=pc\}\cap\{t=tc\}\cap\{match(t,v,vc)\}$ 为真，则称 QoS_i 匹配 $QoSC_i$，记为 *matched*（$QoS,QoSC_i$）。若 $\{p=pc\}\cap\{t=tc\}$ 为真，而 $\{match(t,v,vc)\}$ 为假，则称 QoS_i 与 $QoSC_i$ 矛盾，记为 *conflicted*（$QoS_i,QoSC_i$）。其中函数匹配规则如表 2 所列。

表 2 匹配规则

t，tc	v，vc	return
1	vc > v	FALSE
1	vc ⩽ v	TRUE
0	vc! = v	FALSE
0	vc = v	TRUE
−1	vc < v	FALSE
−1	vc ⩾ v	TRUE

定义 2 如果对于服务订阅 SService 中的每一个 QoS 属性约束，服务 PService 中至少存在一个 QoS 属性与之匹配，且没有任何 QoS 属性发生矛盾，则称 SService 与 PService 匹配，即：$\forall QoSC_i\in SService$，$\exists QoS_i\in PService$，使得 *matched*（$QoS_i,QoSC_i$），且 $\forall QoS_i\in PService$，不存在 QoS_i 使得 *conflicted*（$QoS_i,QoSC_i$），则称 PService 与 SService 匹配，记为 *matched*（*PService*，*SService*）。

3 系统架构

Web Service 发布/订阅系统中有很多角色，在系统中表示为实体，从面向应用的角度，把这些实体按照它们的功能和特点分为 3 类：服务发布者、服务订阅者、服务管理者；这 3 种角色对应了 3 种操作：发布（Publish）、订阅（Subscribe）和通知（Notification）。发布订阅系统可以采用集中式和分布式体系结构，集中式结构扩展性差，在大规模发布订阅系统中容易产生瓶颈。图 1 给出了一个分布式 Web Service 发布订阅系统模型。

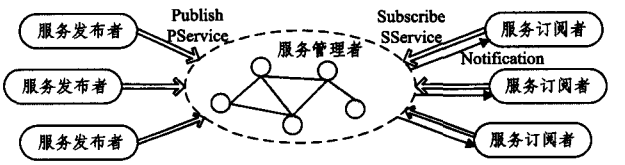

图 1 分布式 Web Service 发布/订阅系统模型

服务发布者和服务订阅者是 Web 服务发布订阅系统中的基本角色。服务发布者向系统中发布（Publish）服务，其每个发布的服务用 QoS 服务模型来描述服务的性能和功能。发布者无需知道订阅者的具体地址信息、订阅者的详细数据，也不关心有多少订阅者收到服务，因此发布者的事件具有“Fire-and-Forget”的特性。服务订阅者即服务的请求者，向系统中订阅（Subscribe）自己感兴趣的服务，服务的订阅用 QoS 订阅模型来描述订阅的要求。同样，订阅者也不关心收到的服务的具体发布者，也无需知道发布者的具体地址及详细数据。

显然发布者和订阅者是松耦合的。

服务发布者和服务订阅者通过发布订阅系统联系在一起。在分布式 Web Service 发布订阅系统中,发布订阅系统由多个服务管理者节点组成。服务管理者收到发布者发布的服务后,将服务保存在本地,将服务的 QoS 属性组织成多级索引结构放入内存,以便随时和订阅进行匹配;订阅者向服务管理者请求订阅服务时,服务管理者负责将订阅与本地保存的服务进行匹配:根据矩形过滤结构减小匹配范围,然后将订阅的 QoS 属性约束与本地保存的多级索引结构进行匹配,如果找到了符合要求的服务,服务管理者则通知(Notification)对应的订阅者,告知其订阅的服务的相关信息;若没有匹配到合适的服务,则通过分布式网络,将订阅分发给网络中其他的服务管理者做同样的操作,若经过处理的订阅失效则停止分发和匹配,服务管理者返回订阅失败。

4 Web 服务匹配算法

4.1 矩形过滤

为了减小服务匹配的范围,提高匹配的效率,首先对服务订阅进行预处理,生成过滤矩阵。

首先生成服务标识 *mark_p* 和服务订阅标识 *mark_s*,用一个字符串来表示,其长度都等于 QoS 模型中属性的数量。过滤矩阵如表 3 所列,每一行表示一个标识的组成元素,每一列表示标识对应的服务是否存在此位置对应的 QoS 属性,若服务中存在相对应的 QoS 属性约束,则 mark 相应的位置为 1,若服务中缺省相对应的 QoS 属性值,则相应的位置为 0,同理生成服务订阅标记,由此生成一个矩形的订阅结构。

表 3 过滤矩阵

	RT	A	T	S	R	…
$mark\_p_1$	1	1	0	1	0	…
$mark\_p_2$	0	1	1	0	1	…
…	…	…	…	…	…	…
$mark\_p_n$	0	0	0	1	1	…
mark_s	1	0	1	1	0	…

4.2 多级索引链表

将发布的 Web 服务生成多级索引表进行保存。首先将所有的服务分解成 QoS 属性的集合,合并重复的属性,按照其属性类型(type)、属性名称(property)和属性值(value)建立索引结构,如图 2 所示。

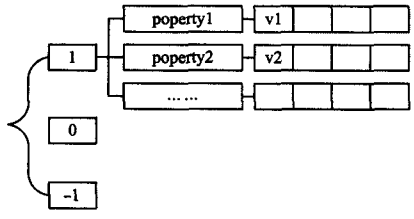

图 2 多级索引结构

多级索引是多个链表的级联,第 1 级是功能属性的 3 大类(1:积极属性;0:中性属性;-1:消极属性)。若 type=1 或 -1,按照 QoS 属性约束的覆盖关系进行排序,使用二分查找法可以迅速查找和匹配索引项,若 QoS 属性约束匹配链表中的某个 QoS 属性,那么该约束以后至链表尾部的约束都匹配该 QoS 属性;若 type=0,则链接的是一个 hash 表,按照其字典顺序排列,每次查找只需在 hash 表中匹配一次。

为了方便处理 QoS 属性和服务之间的隶属关系,生成 QoS 属性到服务 Id 链表的映射,如图 3 所示。每个 QoS 属性对应多个服务 Id,且其中的每个服务 Id 所对应的服务都包含有此 QoS 属性。

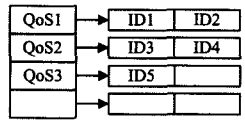

图 3 QoS 属性-服务 Id 链表映射

4.3 服务匹配流程

根据 QoS 和服务订阅模型,基于过滤矩阵可多级索引结构,进行匹配的流程如图 4 所示。

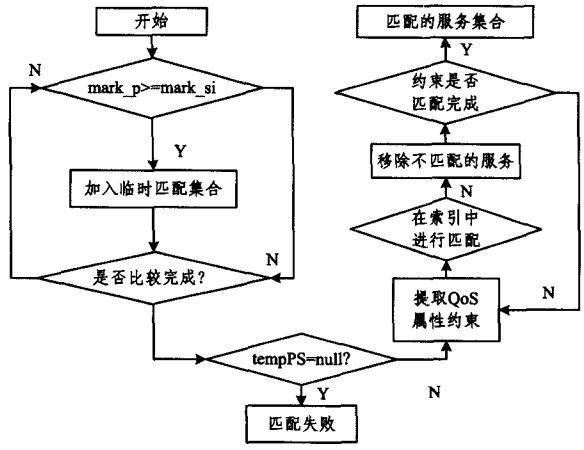

图 4 服务匹配流程

整个服务匹配的步骤如下:

Step1 将服务订阅的 *mark_s* 与过滤矩阵中的 *mark_p* 按整型数据逐个进行比较。若 *mark_p* 大于等于 *mark_s*,则配对成功;否则配对失败。将配对成功的 *mark_p* 对应的服务 Id 加入到临时匹配集合 *tempPS* 中。

Step2 比较完成之后,得到临时匹配集合 *tempPS*,若 *tempPS* 为空,则此次匹配失败,没有找到符合要求的服务;否则进入下一步。

Step3 取出服务订阅的一个 QoS 属性约束,根据其类型、属性名、属性值按顺序在多维索引表中查找匹配的值。首先在索引中匹配类型,例如若操作符为“>”,则匹配积极属性;然后在匹配成功的类型中再根据 QoS 属性约束的属性名继续匹配索引的属性名;匹配成功后,最后匹配属性值,根据覆盖关系可以迅速匹配出不符合要求的服务,并将不符合要求的服务从 *tempPS* 中剔除。然后以同样的方法取出服务订阅的第二个 QoS 属性在索引中进行匹配,以此类推,直到所有约束都匹配完毕。

Step4 匹配完成之后,*tempPS* 即为匹配服务的集合。

5 实验与评价

5.1 实验环境

采用 Java 语言实现了基于 QoS 和多级索引的发布订阅机制,利用 QWS 数据集[16]进行测试。QWS 数据集是一组真实的 Web 服务测试数据,共提供了 2500 多个真实的 Web 服务,并且每个 Web 服务包含 9 个 QoS。除了实现本文所提出的多级索引算法(this)之外,还实现了传统暴力法(Brute)与文献[9]中提出的多维索引和计数法相结合的混合算法(Hy-

brid),并将其融入到 Web 服务发布订阅系统进行对比分析。

实验在操作系统为 Windows7,具有 Core i5 2.6GHz 处理器、4GB 内存的 PC 上完成。Web 服务的发布订阅系统功能正常,能够主动地对服务的 QoS 和订阅的 QoS 约束进行响应。进一步对服务数量和服务订阅数量对算法的影响进行性能分析。因为实际 QWS 数据集只有 2500 多个服务,需要扩展,相应的扩展规则如下:

(1)发布的 Web 服务 QoS 属性与 QWS 数据集相同,在已有 QWS 数据集中收集的服务的基础上,随机增加或减少 Web 服务中的一个或多个 QoS 属性;然后再根据收集到的 QWS 中的 QoS 取值范围随机改变部分 QoS 值,以产生满足实验数量的测试 Web 服务。

(2)在已有 QWS 数据集的基础上,服务订阅者根据 QoS 属性和范围随机生成服务订阅的 SService 的 QoSC 的取值。

5.2 性能分析

(1)服务数量对算法的影响。服务发布者发布 QoS 属性数量为 6～15 的 Web 服务(PService) 10000～640000 个,然后服务订阅者提交一个 QoS 属性约束数量为 9 的服务订阅(SService),服务订阅匹配所需的时间如图 5 所示。

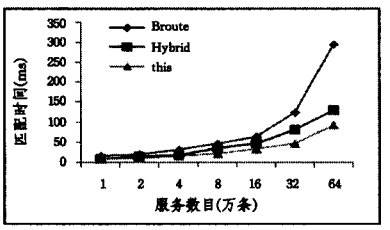

图 5 不同服务数量下服务订阅耗时

从实验结果可以看出,传统暴力法、混合算法和多级索引算法的匹配时间都随着服务数量的增加而增加。当服务数目比较少时,三者效率相差不大;随着服务数量的增多,多级索引算法在匹配时间上要明显短于传统暴力法和混合算法,其在服务数量达到 64 万条时,其匹配时间在 100ms 以内,传统的暴力算法是最慢的,而混合算法中由于运用了计数法,在服务数很大的情况下容易产生瓶颈。

(2)服务订阅数量对算法的影响。针对服务组合的订阅,固定系统中发布的 Web 服务数为 160000,服务订阅者发布属性约束数量为 9 的订阅数为 2～8,测得服务匹配耗时如图 6 所示。

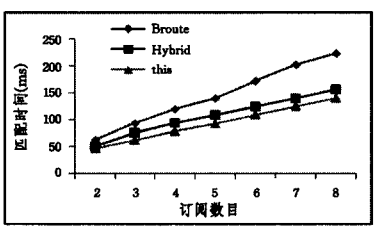

图 6 发布多个订阅所需的耗时

从实验结果可以看出,传统暴力法、混合算法和多级索引算法的匹配时间大致随着服务订阅数量的增加而增加。其中,暴力法效率最低,混合算法效率其次,多级索引算法最好。多级索引法完成订阅匹配 8 条的时耗在 130ms 左右,比传统暴力法快 43%,比混合算法快约 18.7%。

通过服务数量和服务订阅数对算法的影响可以看出,当 Web 服务数越大、服务的订阅数越多时,也就是整个 Web 服务发布订阅系统越繁忙时,本文提出的基于 QoS 和多级索引的 Web 服务发布订阅机制都能够在比较理想的情况下完成服务的发布和订阅,更适应于大规模 Web 服务的管理。

结束语 随着云计算的普及,海量 Web 服务管理将成为一个难题,本文提出了一种基于 QoS 和多级索引的 Web 服务发布订阅机制。引入发布订阅机制有利于实现对大规模 Web 服务的主动管理,利用 Web 服务的 QoS 属性和订阅的属性约束之间的匹配关系形成发布订阅模型。将发布的 Web 服务及其 QoS 和服务订阅一起生成过滤矩阵进行预处理,减小了匹配的范围,利用多级索引技术提高了匹配的速度。实验结果表明,基于 QoS 和多级索引的 Web 服务发布订阅机制比传统方法更能够适用于大规模 Web 服务的管理。

参 考 文 献

[1] Li Qiao, Zheng Xiao. Research Survey of Cloud Computing [J]. Computer Science, 2011, 38(4): 32-37(in Chinese)
李乔,郑啸. 云计算研究现状综述[J]. 计算机科学,2011,38(4):32-37

[2] Duan Q, Yan Y, Vasilakos A V. A survey on service-oriented network virtualization toward convergence of networking and cloud computing[J]. IEEE Transactions on Network and Service Management, 2012, 9(4): 373-392

[3] Zheng Zi-bin, Zhang Yi-lei, Lyu M R. Investigating QoS of Real-World Web Services[J]. IEEE Transactions on Services Computing, 2014, 7(1): 32-39

[4] Yuan Hong-liang, Yin Gang, Wang Huai-min. The Application Integration Framework of supporting publish-subscribe Web services[J]. Computer & Digital Engineering, 2008(3): 81-83, 166(in Chinese)
苑洪亮,尹刚,王怀民. 支持发布订阅的 Web 服务应用集成框架[J]. 计算机与数字工程,2008(3):81-83,166

[5] Zhang Ting-jun. The Design and Implementation of Publish/Subscribe System Based on WSN[D]. Changsha: National University of Defense Technology, 2006(in Chinese)
张庭军. 基于 WSN 的 Web Services 发布/订阅系统的研究和实现[D]. 长沙:国防科学技术大学,2006

[6] Zheng Xiao, Luo Jun-zhou, Cao Jiu-xin, et al. A Publish/Subscribe Based Information Dissemination Model for QoS of Web Services[J]. Journal of Computer Research and Development, 2010(6): 1088-1097(in Chinese)
郑啸,罗军舟,曹玖新,等. 基于发布/订阅机制的 Web 服务 QoS 信息分发模型[J]. 计算机研究与发展,2010(6):1088-1097

[7] Aguilera M K, Strom R E, Sturman D C, et al. Matching events in a content-based subscription system[C]// 18th ACM Symposium on Principles of Distributed Computing (PODC). Atlanta, USA, 1999: 53-61

[8] Campailla A, Chaki S, Clarke E, et al. Efficient filtering in publish-subscribe systems using binary decision diagrams[C]// 23rd International Conference on Software Engineering. IEEE, 2001: 443-452

[9] Xue Tao, Feng Bo-qing, Li Bo, et al. Efficient Matching for Content-Based Publish-Subscribe Systems[J]. Journal of Chinese Computer System, 2006, 3: 529-533(in Chinese)
薛涛,冯博琴,李波,等. 基于内容的发布订阅系统中快速匹配算法的研究[J]. 小型微型计算机系统,2006,3:529-533

[10] Carzaniga A, Rosenblum D S, Wolf A L. Design and evaluation of a wide-area event notification service [J]. ACM Transactions on Computer Systems, 2001, 19(3): 332-383

[11] Altinel M, Franklin M J. Efficient Filtering of XML Documents for Selective Dissemination of Information[C]// 26th International Conference on Very Large Data Bases. 2000: 53-64

[12] Diao Y, Fischer P, Franklin M J, et al. Yfilter: Efficient and scalable filtering of XML documents[C]// 18th International Conference on Data Engineering. IEEE, 2002: 341-342

[13] Zhang Cheng-wen, Su Sen, Chen Jun-liang. Genetic Algorithm on Web Services Selection Supporting QoS [J]. Chinese Journal of Computers, 2006, 29(7): 1029-1037(in Chinese)
张成文,苏森,陈俊亮. 基于遗传算法的 QoS 感知的 Web 服务选择[J]. 计算机学报, 2006, 29(7): 1029-1037

[14] Ma You, Wang Shang-guang, Sun Qi-bo, et al. Web Service Quality Metric Algorithm Employing Objective and Subjective Weight[J]. Journal of Software, 2014, 25(11): 2473-2485(in Chinese)
马友,王尚广,孙其博,等. 一种综合考虑主客观权重的 Web 服务 QoS 度量算法[J]. 软件学报, 2014, 25(11): 2473-2485

[15] Kritikos K, Plexousakis D. Requirements for QoS-Based Web Service Description and Discovery[J]. IEEE Transactions Service Computing, 2009, 2(4): 320-337

[16] Al-Masri E, Mahmoud Q H. Discovering the best Web service [C]// 16th International Conference on World Wide Web. Alberta, CANADA, 2007: 1257-1258

(上接第 101 页)

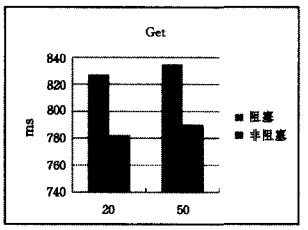

图 11 get 操作在阻塞模式和非阻塞模式下的时间对比

结束语 远程过程调用是 HBase 架构中重要的底层通信框架,随着当前数据服务量的快速增长,客户端的请求越来越频繁,阻塞通信机制已成为影响 HBase 的性能的重要因素。本文分析 HBase RPC 客户端和服务端的通信机制,HBase RPC 客户端的阻塞会导致数据传输性能的下降,为此提出了非阻塞模式的通信模型,并通过 Java NIO 实现了 HBase RPC 非阻塞式客户端。实验分析表明,在大数量及密集数据读写的环境下,HBase RPC 客户端的非阻塞式通信要优于阻塞式通信。

参 考 文 献

[1] NoSQL Database[EB/OL]. 2014. 8. http://nosql-database.org

[2] Apache HBase[EB/OL]. 2014. 8. http://hbase.apache.org

[3] Sun Wei-qin. Java Network Programming [M]. Beijing: Publishing House of Electronics Industry, 2007: 82-86(in Chinese)
孙卫琴. Java 网络编程精解[M]. 北京:电子工业出版社, 2007: 82-86

[4] Chang F, Dean J, Ghemawat S, et al. Bigtable: A distributed storage system for structured data[C]//Proceedings of the Seventh Symposium on Operating System Desgin and Implementation. 2006

[5] Huang Jian, Ouyang Xiang-yong, Jose J, et al. High-Performance Design of HBase with RDMA over InfiniBand[C]// IEEE International Parallel and Distributed Processing Symposium. IEEE, 2012: 774-785

[6] Lu Xiao-yi, Islam N S, Wasi-ur-Rahman M, et al. High-Performance Design of Hadoop RPC with RDMA over InfiniBand[C]// International Conference on Parallel Processing. IEEE, 2013: 641-650

[7] Bao Xian-qiang, Liu Ling, Xiao Nong, et al. HConfig: Resource adaptive fast bulk loading in HBase[C]// International Conference on Collaborative Computing: Networking, Applications and Worksharing. IEEE, 2014: 215-224

[8] Tian Sheng-li, Xu Xi-shan, Yang Shu-qiang, et al. Optimization for the Access Interface of MapReduce in HBase[C]// Collections of Ninth Annual Academic Conference of China Institute of Communications. China Institute of Communications, 2012(in Chinese)
田胜利,徐锡山,杨树强,等. 针对 HBase 的 MapReduce 访问接口的优化[C]//第九届中国通信学会学术年会论文集. 中国通信学会, 2012

[9] Luo Yan-xin. Research and Implementation on HBASE Based Column-Oriented Compression Algorithms [D]. Guangzhou: South China University of Technology, 2011(in Chinese)
罗燕新. 基于 HBASE 的列存储压缩算法的研究与实现[D]. 广州:华南理工大学, 2011

[10] Cheng Peng-sen, An Jun-xiu. The key as dictionary compression method of inverted index table under the HBase database[J]. Journal of Software, 2013, 8(5): 1086-1093

[11] Kang Yi. The Design and Implementation of HBase Large Object Storage[D]. Nanjing: Nanjing University, 2013(in Chinese)
康毅. HBase 大对象存储方案的设计与实现[D]. 南京:南京大学, 2013

[12] Harter T, Borthankur D, Dong Si-ying, et al. Analysis of HDFS Under HBase: A Facebook Messages Case Study[C]// Proceedings of the 12th USENIX Conference on File and Storage Technologies. USENIX, 2014: 199-212

[13] A Summary of Application and Optimization of HBase in Taobao[EB/OL]. 2014. 8. http://blog.nosqlfan.com/html/3694.html(in Chinese)
HBase 在淘宝的应用和优化小结[EB/OL]. 2014. 8. http://blog.nosqlfan.com/html/3694.html

[14] Perfection and Innovation of HBase in Jingdong[EB/OL]. 2014. 8. http://www.sootoo.com/content/455783.shtml(in Chinese)
HBase 在京东的完善与创新[EB/OL]. 2014. 8. http://www.sootoo.com/content/455783.shtml

[15] Liu Shao-hui. HBase Used in Xiaomi [R]. China Hadoop Summit. 2013(in Chinese).
刘绍辉. 小米 Hbase 实践[R]. 中国 Hadoop 技术峰会, 2013

[16] Geoge L. HBase 权威指南[M]. 代志远,刘佳,蒋杰,译. 北京:人民邮电出版社, 2013: 302-304

[17] Hbase Hmaster Architecture[EB/OL]. 2014-8. http://blog.zahoor.in/2012/08/hbase-hmaster-architecture/