

分布式 Web 服务 QoS 注册中的高效负载均衡方法

李 飞 杨放春 苏 森

(北京邮电大学网络与交换国家重点实验室 北京 100876)

摘 要: 该文提出了一种分布式 Web 服务 QoS 注册系统的负载均衡方法。提出了节点负载状态划分的概念。根据负载状态采用不同的负载信息散布策略,极大地减少了网络消耗。提出了基于简单协商的负载均衡方法。负载均衡用数据复制的方式,主要特点是在进行实际负载均衡操作以前即与复制目标节点进行协商,提出合理的复制需求,然后根据对方提供的资源情况,发起复制。提高了负载均衡的效率,降低了复制的盲目性。该方法在分布式 Web 服务 QoS 注册原型系统中进行了实验,达到了较好的效果。

关键词: Web 服务; QoS 注册; 负载均衡

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2009)05-1022-04

Efficient Load-Balance in Distributed QoS Registry for Web Service

Li Fei Yang Fang-chun Su Sen

(State Key Lab. of Networking and Switching, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: A load-balance approach in distributed QoS registry for Web service is proposed. According to peer load status, different load information dissemination policies are applied which significantly reduce network overhead. A load-balance approach based on simple negotiation protocol is proposed. The replication initiator proposes a rational proposal to replication receiver, then it replicates based on opponent's offer. The approach improves efficiency of load-balance and reduces unnecessary replication attempt. The approach is tested on the prototype of distributed QoS registry and proved by experimental result.

Key word: Web service; QoS registry; Load-balance

1 引言

Web 服务质量(Quality of Service, QoS)一般指服务的非功能属性,如响应时间,价格等,是影响用户服务选择和使用体验的关键因素。面向服务体系结构(Service Oriented Architecture, SOA)的基础设施中,必须向服务用户和服务提供商提供中立的,高效的服务 QoS 注册查询机制。传统的服务注册研究,主要关注服务注册信息的描述和查询,对于服务功能描述和 QoS 通常一同存储,一起查询。实际上,服务的功能信息和 QoS 信息的使用方式有着较大区别。功能信息通常比较稳定,用户一次查询结果有较长时间的可用性;而 QoS 的描述参数集虽然比较稳定,但参数值通常会随时间,服务状态等因素不断变化。对于服务的使用者来讲,需要在使用前确定服务当前的 QoS 状态,与服务绑定之后,还需要在整个使用期对 QoS 进行监控。因此,用户对于服务 QoS 信息的使用通常是频繁的查询与更新的过程,尤其是与服务绑定以后,用户没有进行功能查询的需求,但是仍然需要反复访问 QoS 数据。因此, Web 服务的 QoS 信息单独存

储是提高服务注册性能的重要方法。

P2P(Peer-to-Peer)作为一种全新的网络服务模式,近年来得到了很大的发展。基于 P2P 模式的 Web 服务基础设施也已有大量研究,但是主要集中于两个方面:(1)分布式服务注册与发现。主要关注服务的语义描述,以及在 P2P 模式下的叠加网结构和查询方式^[1,2]。(2)分布式服务执行环境。主要研究服务执行协作和调度的模式^[3,4]。分布式的 QoS 注册仅在文献[5]中有所提及,但基于 DHT(分布式哈希表)的查询方式并不适合 QoS 查询。我们已经提出分布式的 Web 服务 QoS 注册系统——Q-Peer^[6,7]。该系统主要特点是,采用 P2P 思想,各节点平等,可以通过在节点间互相复制数据以保持负载均衡,保证整个系统的查询效率。用户必须知道一个服务的功能才会对其 QoS 感兴趣,因此 QoS 检索采用类似 Napster^[8]的方式,将服务功能注册系统作为检索入口,通过服务注册记录,直接访问 QoS 的存储地址。每一 QoS 数据对象有一个主控节点,负责更新 QoS 数据,并且保持多个复制数据的同步。关于 Q-Peer 系统更具体的介绍可以参考文献[6]。

本文对 Q-Peer 中的负载均衡方法进行改进,原有负载均衡方法主要有以下问题:(1)节点间负载信息传播消耗较

2008-02-02 收到, 2008-07-16 改回

国家 973 计划项目(2003CB314806), 国家 863 计划项目(2006AA01Z164)和新世纪人才支持计划(NCET-05-0114)资助课题

大。(2)在 QoS 类复制(Replication of QoS Class, RQC)和 QoS 对象复制(Replication of QoS Object, RQO)中, RQC 方法每次复制一个 QoS 类, 而 RQO 方法每次只复制一个 QoS 对象。两者的性能比较 RQO 方法明显较好, 但 RQO 方法也有显著的问题, 即达到负载均衡的时间较长, 需要的消息数量较大。本文提出根据节点负载状态进行复制准备。并且在复制之前进行执行一个快速的协商协议, 可以在保证复制的效率的同时降低复制过程中的网络消耗, 提高负载均衡的速度。

2 节点状态以及负载信息传播

在文献[6]中介绍了 Q-Peer 负载和容量的基本计算模型。Q-Peer 中一个节点 P 的负载 L 由所有服务 QoS 数据的更新频率 f^u 和查询频率 f^q 相加得到, 见式(1)。节点的容量即是最大可接受的访问频率。

$$L(P) = \sum f^u(Q(s)) + f^q(Q(s)) \quad (1)$$

Q-Peer 的负载信息传播基于维护邻居列表的方法。Q-Peer 各节点之间通过交换剩余容量信息来寻找有空闲容量的邻居节点以保证在需要进行复制的时候, 可以找到可用的邻居。

实际上, 交换信息的目的是为了准备复制, 因此, 系统中各节点的容量信息并不需要无条件的在任何时候都与邻居节点交换, 尤其是在负载较低的情况下。假设节点 P 的容量为 C , 高负载阈值控制变量为 $a(0 < a < 1)$, 节点负载为 L 。节点分为 3 种状态: 低负载状态 $L < aC$, 高负载状态 $aC < L < C$, 过负载状态 $L > C$ 。低负载状态可以接受复制, 而高负载状态不接受, 过负载状态的时候会发起复制。

本文提出, 根据节点负载状态的不同, 采用不同的邻居列表更新策略。

低负载状态下, 以较低的频率更新邻居列表, 同时确认邻居是否在线。

在高负载状态和过负载状态下, 节点需要准备邻居列表。节点以较高频率周期性地更新邻居负载信息, 并保持邻居列表按照空闲容量降序排列。如果出现邻居列表中第一个邻居空闲容量较低或者邻居总数较低的情况, 则启动随机游走寻找新的邻居。

上述负载信息传播方法显著地改善了由于负载信息传播导致的消息数量, 并且保持了负载信息的有效性, 使得节点在需要数据复制的时候仍然可以立即找到可用的邻居。

3 基于简单协商的负载均衡方法

3.1 高负载状态下的复制准备

本文提出的复制方法仍然以 QoS 对象为基本元素。高负载状态下的复制准备主要是准备备选的 QoS 复制对象集合。每一节点实时地对自己所有的 QoS 对象进行监控, 记录单个 QoS 对象的访问频率。每一 QoS 对象已有的副本数目在进行 QoS 数据同步的时候由主控节点告知。

复制集的准备并不是以访问量为标准, 本文用复制效用 U 来表示一个 QoS 对象复制的价值。当 QoS 对象有多个副本的时候, 复制访问量最大的对象并不一定最大限度地减少当前节点的访问量。效用就是表示复制某一 QoS 对象以后, 节点减少的访问量。 s_i 是一个服务, $Q(s_i)$ 为其 QoS 数据对象, 而 r_i 为当前已有的 QoS 副本数, 包括原始数据, 见公式(2)。

$$U(Q(s_i)) = \frac{f^q(Q(s_i))}{r_i} - \frac{f^q(Q(s_i))}{r_i + 1} \quad (2)$$

定义备选复制集合为 $Q_{cand} = \{Q(s_1), Q(s_2), \dots, Q(s_n)\}$, 则 $U(Q_{cand}) = \sum U(Q(s_i))$, 其中 $Q(s_i) \in Q_{cand}$ 。 Q_{cand} 满足条件式(3)。即依降序选取效用最大的 QoS 对象, 使得整个集合的复制效用恰好达到使当前节点由高负载状态进入低负载状态。

$$\left. \begin{aligned} &\forall Q(s) \in Q_{cand}, \forall Q(s') \notin Q_{cand}, U(Q(s)) > U(Q(s')) \\ &U(Q_{cand}) > (1 - \alpha)C \\ &\forall Q(s) \in Q_{cand}, U(Q_{cand} - Q(s)) < (1 - \alpha)C \end{aligned} \right\} \quad (3)$$

由于互联网中的服务访问通常服从 Zipf 分布^[9], 即小部分的服务占有了大部分的访问量, 备选集合包含的 QoS 对象数目相对于整个节点的 QoS 对象数目通常非常小。因此备选复制集合可以相当高效地维护。

3.2 过负载状态下的数据复制

本文提出一个单次交互的复制前协商协议。这样的协议可以使复制发起者在复制以前确认容量, 复制接收者可以进行自己的负载控制, 寻找数据重复, 并且预留资源。由于数据复制发生在节点过负载的时候, 协商协议必须简单快速。

在容量协商协议中, 复制发起方首先提出一个提议(Proposal), 内容包括容量需求 req_cap 和请求复制的集合 req_Q 。由于系统经过一段时间的运行之后部分 QoS 对象可能已有副本, 所以必须在复制以前确认目标节点是否已包含准备复制的对象, req_Q 即为待复制的 QoS 标识集合。可能接收复制的邻居节点返回可提供的资源(Offer), 包括允许复制的容量 off_cap 和允许复制的集合 off_Q , 容量为 0 则表示拒绝复制。此时接收方已将这部分资源预留。发起方不再确认, 直接开始复制。如果发起方没有立即开始复制, 接收方资源预留一个短后即取消预留的容量, 视为发起方放弃复制。见图 1。

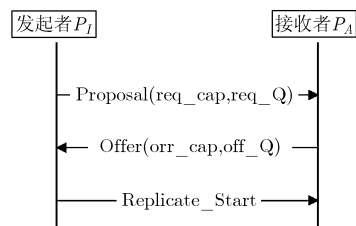


图 1 复制协商协议

容量需求 req_cap 总是等于复制 Q_{cand} 以后 Q_{cand} 在接收节点上产生的负载。一个 QoS 集合 QS 产生的复制负载 f^r 定义如式(4)，因此 $\text{req_cap} = f^r(Q_{\text{cand}})$ 。请求复制集合 $\text{req_}Q = Q_{\text{cand}}$ 。

$$f^r(QS) = \sum_{Q(s_i) \in QS} \left(\frac{f^a(Q(s_i))}{r_i + 1} + f^u(Q(s_i)) \right) \quad (4)$$

实际准备复制的集合根据对方提供的 Offer 产生。定义复制集合为 Q_{rep} ， Q_{rep} 满足式(5)所示的条件。即从 Q_{cand} 中按照复制效用的降序依次选取 QoS 对象，使得复制负载尽量接近对方提供的复制容量，并且对方没有包含该对象的副本。

$$\left. \begin{aligned} &Q_{\text{rep}} \subset Q_{\text{cand}} \\ &\forall Q(s) \in Q_{\text{rep}}, Q(s) \notin \text{off_}Q \\ &f^r(Q_{\text{rep}}) < C_{\text{offer}} \\ &\forall Q(s) \in Q_{\text{rep}}, \forall Q(s') \in Q_{\text{cand}} - Q_{\text{rep}}, U(Q(s)) > U(Q(s')) \end{aligned} \right\} (5)$$

一次复制以后，节点首先将刚才用到的邻居删除，因为如果需要再次发起复制，该邻居可能会因为已经包含了大多需要复制的对象而不可用，即使剩余容量还很大。然后重新生成 Q_{cand} ，根据当前的邻居列表状况，重复执行复制协议。

4 实验结果

实验在 Q-Peer 原型系统上进行，该原型系统为部署在 2 台 Linux 服务器上的多个进程，可以模拟多个节点同时自治运行的情况。

实验主要有以下几个测试目标：(1)基于状态的负载信息交换的性能。(2)基于协商协议的复制效果。(3)协商协议的性能。测试一共有 50 个节点，每一节点的最大允许的容量为 500-10000(每分钟访问次数)之间的随机数。负载控制变量 α 为 0.8。QoS 数据有服务提供者测试程序添加。每一提供者每分钟更新 QoS 一次。QoS 查询请求服从 Zipf 分布，即少数 QoS 有很高的访问频率。这一分布比较接近真实的网络访问特征。大约 50k 条 QoS 数据记录随机分布在节点中。系统的查询请求从 50k/min 逐步增长至 190k/min。邻居列表长度为 8，随机游走最大允许 10 步。所有实验均是取 100 次重复的平均值。

图 2 显示了负载更新消息的数量变化。周期性的更新方法，每个节点大约每分钟更新邻居 12 次，每次都需要向所有邻居询问。随机游走消息也作为一种负载更新消息计算在内。而基于状态的更新方法，在系统低负载的时候，大约每分钟更新邻居一次。高负载的时候每分钟更新 12 次。从图中可以看到，周期性的更新，到整个系统中的负载更新消息已开始就在一个较高水平上，随着负载增加，由于随机游走的增加，更新消息继续增加。而基于状态的更新则使得负载更新消息从较低水平逐渐随系统需要缓慢增长。

我们假设节点在过负载的时候会简单的将多余请求丢弃，图 3 显示了两种复制方法可以达到非常相近的复制效果。

鉴于篇幅原因，整体系统平衡度等其余参数情况此处略去，基本与逐对象复制相同，请参考文献[6]。基于简单协商的复制主要目标是改善复制的效率。从图 4 可以看到，基于简单协商的复制，在达到同样复制效果的情况下，平均复制次数显著减少，从而提高了负载均衡的速度并且减少了复制中的消息开销。

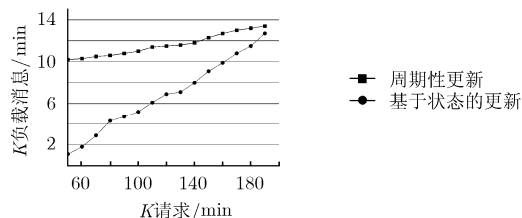


图2 负载更新消息数量

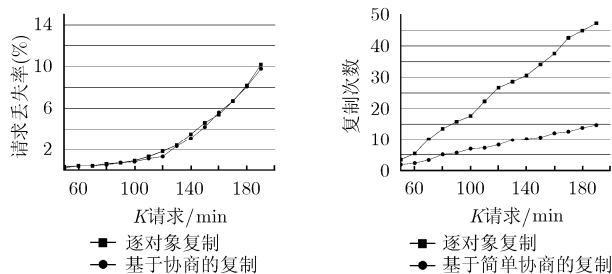


图3 复制效果

图4 复制次数

5 结束语

QoS 存储与查询是 Web 服务研究中的重要问题。基于我们曾经提出的 Q-Peer 分布式 QoS 注册系统，通过对节点状态进行区分和维护，确定了不同状态下节点的功能模式。提出了一种新的负载信息散布方法，有效地降低了负载均衡相关的消息数量，即网络消耗。提出了一种数据适度复制的负载均衡方法。该方法在复制前与复制目标节点进行简单协商，可以保证复制的效果，同时又减少了试探的过程。实验证明本文提出的负载均衡方法有较好的效果。该方法也可以应用于其他类似的无结构 P2P 网络。

参考文献

- [1] Koloniari G and Pitoura E. Peer-to-peer management of XML data : Issues and research challenges. *ACM SIGMOD Record*, 2005, 34(2): 6-17.
- [2] Schmidt C and Parashar M. A peer-to-peer approach to Web service discovery. *Proceedings of the 13th International Conference on World Wide Web*, New York, NY, USA (2004) 211-229.
- [3] Chafle G, Chandra S, and Kankar P, *et al.* Handling faults in decentralized orchestration of composite Web services. *Proceedings of the 3rd International Conference on Service Oriented Computing, ICSOC'05*, 2005, LNCS 3826: 410-423.

- [4] Benatallah B, Sheng Q, and Dumas M. The Self-serv environment for Web services composition. *Internet Computing*, 2003, 7(1): 40-48.
- [5] Gibelin N and Makpangou M. Efficient and transparent Web-services selection. Proceedings of the 3th International Conference on Service Oriented Computing, 2005, LNCS 3826: 527-532.
- [6] Li F, Yang F C, and Shuang K, *et al.* Q-Peer: A decentralized QoS registry architecture for Web services. Proceedings of International Conference on Service-Oriented Computing (ICSOC), 2007, LNCS 4749: 145-156.
- [7] LI F, Yang F C, and Shuang K, *et al.* Peer-to-Peer based QoS registry architecture for Web services. Proceedings of the 7th IFIP International Conference on Distributed Applications and Interoperable Systems, DAIS'07, 2007, LNCS 4531: 133-138.
- [8] Napster Homepage, <http://www.napster.com>.
- [9] Adamic L A and Huberman B A. Zipf's Law and the Internet, *Glottometrics*, 3, 2002: 143-150.
- 李 飞: 男, 1980 年生, 博士生, 研究方向为下一代网络, 新一代互联网服务.
- 杨放春: 男, 1957 年生, 教授, 博士生导师, 主要研究方向为通信软件、智能网、下一代网络.
- 苏 森: 男, 教授, 博士生导师, 主要研究方向为下一代网络、新一代互联网服务.