

一种 Web 集群系统下的 QoS 控制策略

杨 武^{1,2} 李 波² 李双庆¹

(重庆大学计算机科学与工程学院 重庆 400044)¹ (重庆工学院计算机学院 重庆 400050)²

摘 要 本文提出了一种在 Web 集群环境下的 QoS 控制策略。集群前端分配器以会话单位来分配访问请求,保证后端服务器间负载的合理均衡分配,同时后端服务器采用基于会话的 QoS 控制策略,实现对集群系统的服务质量控制。

关键词 Web 集群, Web QoS, 负载均衡, Session

A Web QoS Strategy in Web Cluster System

YANG Wu^{1,2} LI Bo² LI Shuang-Qing¹

(College of Computer Science and Engineering, Chongqing University, Chongqing 400044)¹

(College of Computer Science and Technology, Chongqing Institute of Technology, Chongqing 400050)²

Abstract In this paper, a Web QoS strategy in the cluster system is proposed. In the front-end dispatcher, the HTTP requests in a session are dispatched to a back-end server as a whole to ensure the load balance. In the back-end servers, a session-based Web QoS strategy is imposed.

Keywords Web cluster, Web QoS, Load balance, Session

1 引言

Web 服务器集群通过一定的技术方法,将若干台 Web 服务器以一定的方式组织起来,构成一台逻辑上单一的、功能更强的 Web 服务器,从而获得比单一 Web 服务器系统更大的系统容量和处理能力。如果 Web 服务器集群不支持 QoS 控制,在系统重载或过载的情况下,Web 集群系统的处理能力将不能充分发挥,而目前绝大多数的 Web 服务器集群都不支持 QoS。

Web 服务器集群的 QoS 控制比单一 Web 服务器的 QoS 控制更加复杂和困难^[5],系统不仅需要保证每台服务器的资源分配和管理满足 QoS 的要求,而且必须从集群系统的角度来协调每台服务器上的资源分配,最大限度地发挥集群系统的能力,满足 Web 应用的 QoS 需求。

Web 服务器集群系统的 QoS 控制一般需要从集群系统和 Web 服务器两级分别进行控制。首先,集群系统级确定 HTTP 请求的 QoS 需求,根据负载均衡、访问局部性和 QoS 控制的策略,决定请求在后端服务器上的分配,后端服务器负责对分配到的请求实施有效的 QoS 控制。

因此,针对 Web 服务器集群系统的 Web QoS 控制问题,论文提出了一种基于会话 Session 的 Web QoS 控制策略,该策略能保证后端服务器间负载的合理均衡并实现 Web 应用的 QoS,模拟实验验证了该策略的可行性。

2 Web 服务器集群的 QoS 控制策略

第 7 层(L7)交换技术与 Web 集群系统的 QoS 技术有密切关联,L7 交换技术是进行 QoS 控制的基础。因为只有根据用户的 HTTP 请求的内容,才能识别用户相关信息(如 URL、COOKIE 等)并实现对用户的分类服务。由于在 Web 服务器集群系统中实现 QoS,与常规的负载分配策略有时相互制约,因此在考虑负载分配策略时,优先遵循 QoS 的控制思想,同时兼顾负载均衡^[1]。

2.1 集群前端分配器的分配策略

HTTP 协议是无状态的协议,每个 HTTP 请求是独立,与其他的请求没有相关性。在电子商务中,一个处理流程往往需要很多个请求按顺序共同来完成,这就需要记录每个请求所在的状态。解决这个问题的最常用的方法是用 Session 技术。Session 是利用 HTTP 协议的 Cookie 来实现的,客户端在进行第一次请求时,服务器会生成 Cookie 值,这个值也就是服务器在服务器端为客户端建的一个索引文件,该文件保存着客户端的信息,如:用户在 Session 所定义的全局变量,然后再将 SessionId 作为一个 Cookie 保存在用户端的文件系统中。当服务器响应客户端的第一次的请求时,会把这个 Cookie 发给客户端并保存在客户端的临时文件(Cookie 文件)中。以后客户端浏览器每次发出请求时,都会在 HTTP 头中带上这个 Cookie 值,服务器就会根据这个 Cookie 值,取得保存在服务器上的内容。因此,集群前端分配器在对请求进行分配时,就必须把同一个 Session 的请求发送到同一个服务器上,这样服务器才能正确读取该请求所在的 Session 的值。否则,如果同一个 Session 的请求发到不同的服务器上,就会出错。

集群前端分配器的请求分配算法必须确保同一个 Session 的 HTTP 请求发送到同一个后端的服务器上,为此本文实现的集群前端分配器的请求分配策略,就是把 Session 作为不可分割的原子单位来进行处理,保证同一个 Session 的请求由同一个服务器来处理。

本文实现的集群前端分配器的请求分配策略主要思想是,当前端分配器与客户端建立 TCP 连接,收到 HTTP 请求后,首先读取 HTTP 头中的 CookieId 后的数字。通过该数字计算出一个哈希值,用该哈希值对后端服务器的台数进行取模运算,所得结果就对应于后端服务器的编号,系统将此请求负载分配到该编号的服务器上。由于客户端在每次 HTTP 会话过程中 CookieId 不会发生变化,因此分配器计算出的哈希值也不会变化,这样就保证了同一 HTTP 会话中的 IP 包发送到了同一台服务器上。

*)项目资助:重庆市科技攻关项目:编号:2001.6715。杨 武 副教授,博士研究生,主要研究方向:分布式处理,计算机网络。

由于集群前端分配器的请求分配算法好坏,对系统性能的影响很大。所以请求分配算法不能因为追求理想的负载均衡而不考虑算法本身的运行开销。本文的请求分配策略时间和空间复杂度都很小,同时,前端分配器不需要动态维护负载与服务器的映射表,因此降低了系统的处理开销。另外 Web 集群系统采用 TCP Handoff 技术^[3],提高了交换的效率。本文请求分配策略要求后端服务器能够为所有可能的请求服务,所以后端各个服务器的内容是全镜像的。

2.2 集群后端服务器的 Web QoS 控制策略

目前,单一 Web 服务器实现基于 Session 的访问控制策略之一是 HP 公司 Cherkasova 提出的一种方法,该方法的主要思想是,在服务器超载的情况下,能够保证较长的 Session 与较短的 Session 一样拥有同等的被服务器完成的机会。统计表明在未采用控制策略时,如果系统超载,能够顺利完成的通常都是较短的 Session,而请求数目较多的那些长 Session 由于拒绝服务而中断的机率较大,因此存在对较长的 Session 的歧视情况。

在 Web 集群下实现 QoS,就要对不同级别的用户提供不同的服务质量保证。Cherkasova 提出一种方法仅仅是系统的访问控制,没有实现分类服务。本文采用一种改进的访问控制策略来实现 QoS 控制,通过调节不同级别的用户的访问控制策略的控制参数,达到为不同等级用户提供不同服务质量保证的目的。Web 集群后端服务器采用本文的访问控制策略后,如果后端服务器的负载到达一定的负载量,就对低级别用户采取较严格的访问控制策略,限制其能够获得的系统资源;而对较高级用户则不采用或采用较弱的控制策略,仍然接受新的 Session 请求。如果负载继续加重,就依据用户级别从低到高,对用户进行访问控制,从而实现多不同级别实现分类服务,保证 QoS 的实现。

3 模拟实验

首先为 Web 服务器集群建立负载模型,该模型参照了文[1]的负载模型。模拟实验以 4 台服务器为模拟环境。请求负载和服务器集群的处理能力采用无量纲的数值表示,假定每

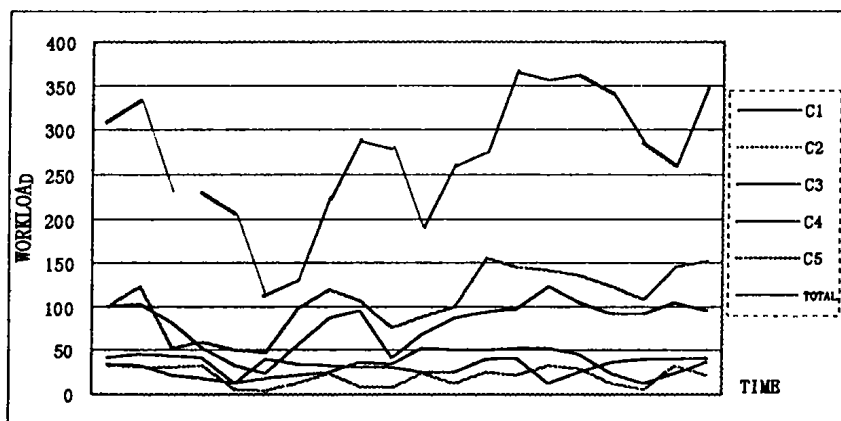


图 1 Web 服务器集群模拟请求负载

个服务器的处理能力是 200。

模拟实验是以 24 小时为一个周期,以 5 分钟为单位时间间隔来描述系统。用户分为 C1, C2, C3, C4 和 C5 等五类,级别从高到低。图 1 为 Web 服务器集群模拟请求负载,图 1 反映了这些用户在不同时间段的负载情况及集群系统的总负载情况。表 1 为该负载模型的特征参数。

表 1 请求负载模型特征系数

类别	C1	C2	C3	C4	C5	总计
最大值	53	34	45	126	156	365
最小值	0	2	10	23	47	112
平均偏差	11.8	9.14	7.68	21.65	28.6	58.14
平均值	32.6	20.5	33.85	81.75	106.2	268.6

为了更清楚地掌握 Web QoS 控制策略的效果,首先对图 1 描述的负载模型采用无 QoS 控制的策略进行实验,获得的结果如图 2 所示。其中波形低表示该时间段无请求被丢弃,波形高表示出现了请求丢弃。可以看出,由于没有对各种服务类加以区分,当系统负载量超过所能处理的上限时,各种服务类被平等处理,均出现了请求负载被丢弃的情况。

图 3 是采用论文提出的 Web QoS 控制模型后,Web 服务器集群系统的响应情况。可以看出,系统优先满足了级别高的用户的资源需要,级别最高的 C1 绝大部分时间的服务都得到了保证;C2 在整个实验中的 Web QoS 均得到了保证;级别较高 C2 和 C3 大部分时间里的资源需求都得到满足,只有很少的时间段出现了资源不能满足的情况;级别最低的 C4 和 C5 系统资源紧张时,频繁出现请求被丢弃的现象。仿真实验的结果,与我们的预期效果完全一致。

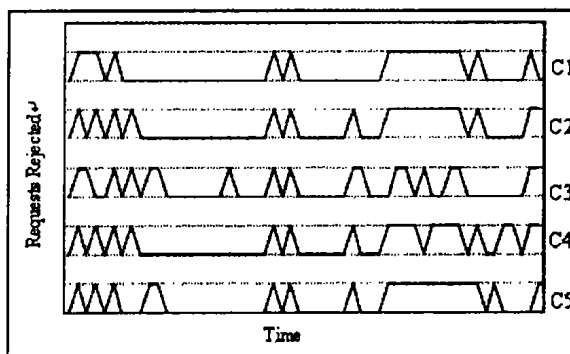


图 2 未采用 QoS 控制时服务器拒绝请求的情况

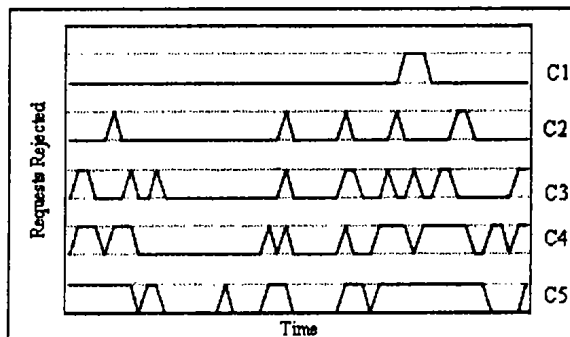


图 3 采用 QoS 控制时候服务器拒绝请求的情况

小结 Web 服务器集群技术是解决服务器性能问题的关键技术,但在现有的 Web 服务器集群系统中,普遍没有考

虑 Web 应用的 QoS 问题。Web 服务器集群环境下的 Web QoS 控制更是一个崭新的研究领域,它不仅要面临单一 Web 服务器环境下 Web QoS 控制的问题,而且需要从集群系统的角度来合理调度资源,满足应用对 Web QoS 的要求。

本文根据 Web 应用中 Session 的 HTTP 请求要求访问服务器的一致性,提出了在集群系统的分配器上采用基于 SessionID 的分配算法。后端的服务器采用改进访问控制策略。模拟实验结果表明,该控制模型对 Web 服务器集群系统的 QoS 具有比较满意的控制能力。

参考文献

1 Cherkasova L, Phaal P. Session Based Admission Control: a Mechanism for Improving the Performance of an Overhead Web Server. [HP Laboratories Report No. HPL-98-119]. June, 1998.

Http://www. hpl. hp. com/98/HPL-98-119. html 8/HPL-98-119. html
 2 李波,李双庆,程代杰. Web 服务器集群中 TCP Handoff 技术及其实现. 计算机工程与应用. 2004
 3 Valeria C, Michele C, Philip S Y. Dynamic Load Balancing on Web-Server Systems. IEEE Internet Computing, 1999(5/6):28~39
 4 Nichols K, et al. Definition of the Differentiated Service Filed in the IPv4 and IP v6 Headers. RFC 2474, Network Working Group
 5 Mohit A, Darren S, Peter D, Willy Z. Scalable Content-aware Request Distribution in Cluster-Based Network Servers. In: Proc. of the 2000 Annual Usenix Technical Conference, San Diego, CA, June 2000
 6 Vivek S, Mohit A. Locality-Aware Request Distribution in Cluster-based Network Servers. In: Proc. of ASPLOS-VIII, ACM SIGPLAN, 1998. 205~216
 7 Teo Y M, Ayani R. Comparison of Load Balancing Strategies on Cluster-based Web Servers. Transactions of the Society for Modeling and Simulation, 2001

(上接第 4 页)

vocation 调用传递给 EJB 容器,进行实际的 EJB Home 接口的调用。方法调用执行完成,将调用结果或产生的异常按照

原来路线返回客户端。在 Object 伺服器中的处理过程与 Home 伺服器类似,不再详细论述^[9]。

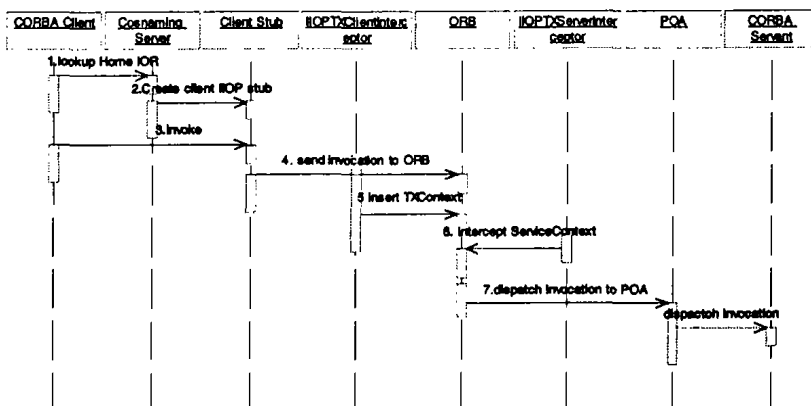


图 7 客户端发出调用后的处理过程

4.3 CORBA 事务服务的实现

为了给没有事务管理器的客户端提供一个划分事务边界和分布事务管理的接口,以将所有事务处理工作委托给服务器端处理,在系统中实现了 CORBA 事务服务,响应 CORBA 客户端对应用服务器的事务服务的调用^[9]。CORBA 事务服务的实现以应用服务器内置的事务管理器为基础。

注册在事务服务 POA 中的 CORBA 事务伺服器负责提供实际的事务服务调用。CORBA 事务伺服器,提供了调用事务服务的一些标准操作,如: create_transaction, create_sub-Transaction, rollback, commit, get_Coordinator, get_terminator 等。

创建 CORBA 事务对象的过程如下:

(1)获得 ORB 服务的实例。(2)通过 ORB 获得 Root-POA。(3)为事务服务生成具有持久生命周期策略的事务服务 POA。(4)创建事务服务伺服器,得到事务工厂的 CORBA 对象引用。(5)将事务工厂的 CORBA 对象引用注册到 Cosnaming 名字服务器中,以供调用。

结束语 本文主要阐述了为支持分布式事务环境中基于 IIOP 协议访问应用服务器端的 EJB 组件所给出的一种解决方案,它解决了如下问题:将客户端请求转发到服务器端的 EJB,实现事务上下文在客户端和服务端端的传播,并且借助应用服务器的事务服务功能,为没有事务管理器的 CORBA 客户端提供了进行分布事务管理的 CORBA 事务服务。本文的工作主要有以下几个特点:(1)基于 JMX 规范的可扩展管

理框架,使整个系统的模块化程度比较高,各个模块具有良好的可配置性,能够方便地替换 ORB 实现、Cosnaming 名字服务器、事务管理等。(2)基于可移植请求拦截器传递事务上下文,使处理简单灵活,而且实现的服务具有在不同 ORB 实现之间的可移植性。(3)以流式方式实现的 CORBA 伺服器,使事务互操作的处理开销较少。本文讨论的工作已经在中国科学院软件研究所研发的 J2EE 应用服务器 OnceAS 中得到了实现。在本文工作的基础上,进一步的工作是要考虑在分布式环境中通过 IIOP 协议对 EJB 组件进行调用时的安全因素,即调用过程中事务上下文的安全传播及在服务端的验证等工作。

参考文献

1 Sun Microsystems. Java 2 Platform Enterprise Edition Specification, v1. 3. March 2001
 2 Sun Microsystems. Enterprise JavaBeans Specification, Version 2. 1. June 2003
 3 Sun Microsystems. JMX Reference Implementation 1. 2. 1. Aug. 2003
 4 OMG. The Common Object Request Broker: Architecture and Specification, version 2. 3. 1, Oct. 1999
 5 范国闻. J2EE 应用服务器关键技术研究: [博士学位论文]. 中科院软件所, 2004
 6 范国闻, 等. WebFrame: 一种多层次可扩展的 Web 应用服务器. 计算机学报, 2004, 27(4): 451~460
 7 OMG. Java Language to IDL Mapping Specification, Version 1. 1, June 2001
 8 OMG. Transaction Service Specification, Version 1. 2, May 2001
 9 http://java. sun. com/j2se/1. 4. 2/docs/guide/rmi-iiop/