

基于区分 Web QoS 的负载均衡集群模型^①

张俊星, 马建红, 周松松

(河北工业大学 计算机科学与软件学院, 天津 300401)

摘要: 随着电子商务的应用逐步深入, 用户访问量的激增且服务请求多样. 如何实现对所有请求的快速响应是当前解决的问题. 针对此问题, 本文提出采用基于区分 Web QoS 的负载均衡技术. 建立了基于区分 Web QoS 的负载均衡的集群模型. 根据请求类型和用户权限划分服务等级, 高服务的请求具有高优先调度权, 在集群当中通过动态反馈技术均衡调度到某个节点, 从而达到区分 Web QoS 服务的目的, 同时也保证集群服务器的负载均衡. 该模型在网络环境及硬件环境相同的条件下, 与常用的负载均衡技术进行了比较, 实验结果证明本文提出的方法效果显著.

关键词: 区分服务; 服务等级; 动态反馈; 负载均衡;

Web QoS Distinction Based Load Balancing Cluster Model

ZHANG Jun-Xing, MA Jian-Hong, ZHOU Song-Song

(School of Computer Science and Software Hebei University of Technology, Tianjin 300401, China)

Abstract: With the gradual deepening of e-commerce applications, the user accesses are sharply increased and the service requests become more diverse. How to achieve quick response to all requests is the current problem. To solve this problem, this paper proposes a Web QoS distinction based load balancing technology. Establish Web QoS distinction based load balancing cluster model. According to the type of service request and the service level for the division of user authority, high level service requests with high priority scheduling rights, which in the cluster are scheduled to a node by dynamically balancing scheduling feedback technology, so as to achieve the purpose of distinction of the Web QoS services, meanwhile, also to ensure the load balance for the cluster server. Under the same network and hardware environment conditions, this load balancing cluster model has been tested and compared with some commonly-used load balancing technologies, the experimental results proved that the proposed method was much more effective.

Key words: Web QoS distinction; service level; dynamically feedback; load balance

随着 Internet 的迅猛发展, 随着 Internet 的发展, 网络数据量的激增. 这对 Web 服务器的处理能力提出了严峻的挑战. 在 Web 服务器过载的情况下, 具备端到端的网络层 QoS 控制保证的高级流仍可能遭受 Web 服务器的拒绝, 由此可见, 服务器从某种程度上说成了实现端到端的 QoS 的瓶颈. 没有好的服务质量的服务已经不能承受大量的用户请求. 集群服务器以其高可扩展性、高可靠性和高性价比, 为 Web 服务器系统带来了新的解决方案^[1].

HP 实验室的 Bhatti 和 Friedrich 在 Apache 服务器

的基础上引入了 Web QoS 控制机制, 具体做法是增加了一个连接管理进程处理到达的请求, 根据 HTTP 请求的类别将它们放入不同优先级的队列当中. 系统将优先级高的队列中的请求分配给工作进程, 使它获得更好的服务质量. 在 Web 服务器集群系统中实现 QoS 控制比单个 Web 服务器中控制更加复杂和困难, 系统不仅要保证每台服务器的负载均衡, 同时要做到对后端服务器资源从 QoS 控制的角度进行调度和分配.

总之, 在 Web 服务器集群系统中实现 QoS 控制

^① 收稿时间:2013-07-14;收到修改稿时间:2013-08-26

机制与策略,为不同类型的用户或请求提供性能保证和区分的服务,这是目前 Web 发展迫切需要解决的问题.

1 相关工作

现有的 Web QoS 的方法有很多,例如:动态网页静态 cache 的方法,这种方法减少了动态网页每次执行对服务器带来的负荷.但却不适用于实时交互型强的应用,如在线交易. 优先调度及提供专用资源的方法,该方法可以有效地改善高优先级请求的延迟性能,同时对低优先级的请求影响也比较小. 集中控制和资源分配的方法,该方法是为不同的页面请求设置不同的优先级,并相应的分配服务器资源. 并支持服务器资源的预订. 改进 Web 服务器应用软件的方法,该方法可以有效和准确地实现所希望的 Web QoS 的区分,但是由于这种方法需要对 Web 服务器软件进行一定程度的修改,所以在实际应用中存在着通用性和可扩展性等方面的局限. 网络 QoS 和 Web 服务器 QoS 的整合方法. 该方法综合考虑网络和 Web 服务器的处理延迟,尤其适合于延迟敏感的应用程序^[2].

现如今负载均衡的算法大体上可以分为两类. 一类是静态的负载均衡算法,不考虑各个服务器的状态. 如基于 DNS 的论转法、加权论转法、散列值法,此类方法不考虑后端服务器的性能及实时的负载状况,这

样很容易造成服务器的负载倾斜;另一类是动态的负载均衡算法,如最小连接数法、加权最小连接调度算法、负载最小调度算法,此类方法可以根据各节点服务器的实际情况动态分发用户请求,近几年的研究工作主要是针对动态负载均衡算法的研究^[3].

在深入了解了区分 Web QoS 的方法和负载均衡的一些算法之后,本文后续部分提出建立一种基于区分 Web QoS 的负载均衡集群模型. 来实现对用户请求的区分服务,并保证集群整体的高性能. 该模型同时兼顾了,对私有数据的保护,对事务型数据缓存,对服务器节点过载迁移的冗余技术,对服务器间请求转发会话保持技术.

2 基于区分服务的负载均衡模型

基于区分服务的负载均衡模型是结合了服务区分和动态反馈的一种负载均衡策略模型. 模型结构图如图 1 所示,请求流进入负载均衡器,首先进行请求的分类,其次一方面根据请求的分类和用户的权限进行服务等级的划分,另一方面使用动态反馈策略,实时的收集后端服务器的的负载信息,最后再进行请求的调度和分配. 对于私有数据的请求,通过反向代理机处理. 各台服务器做好冗余处理,配备相应的备份服务器. 请求再服务器间进行迁移时,要做好会话保持. 以下是对各个部分进行详细介绍.

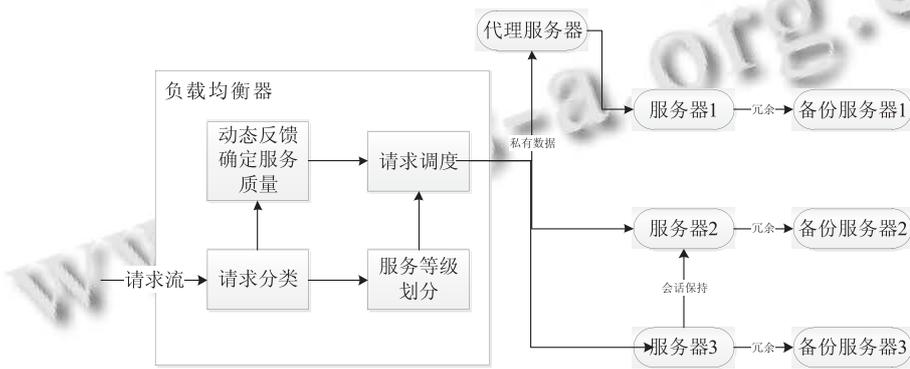


图 1 模型结构图

2.1 请求内容分类

网页一般分为静态页面和动态页面. 从用户角度又可分为私有数据和公有数据. 动态页面由于要频繁的查询数据库,因此对服务质量的要求较高. 而私有数据又需要采取加密保护措施. 这就使得不同的请求

任务消耗的系统资源也会差异很大而且服务器采取的策略也不相同. 借鉴 CAP(Client Aware Policy)^[4]的做法并对其进行扩展,将 WEB 服务请求分为 4 种类型:

1) 发布型: 主要是指静态信息(如 HTML 页面及嵌入的对象),以及轻微动态网络发布服务.

2) 事务型: 结果来自动态数据库查询, 且查询条件一般是由用户通过表单提交得到. 它通常需要进行密集的磁盘访问, 因此也称磁盘密集型服务.

3) 私有数据: 涉及商业机密或者个人隐私的数据.

4) 共有数据: 相对于私有数据而言, 不涉及机密和隐私的数据.

对于私有数据, 集群要防止恶意攻击, 保证私有数据的安全. 本文采用反向代理技术. 当客户机向站点提出请求时, 请求将转到代理服务器. 然后代理服务器将请求发送到内容服务器. 这样, 代理服务器就在安全数据库和可能的恶意攻击之间提供了一道屏障. 就算是攻击成功, 作恶者充其量也仅限于访问单个事务中所涉及的信息, 并不能攻击到真正的内容服务器^[5].

对于事务型数据, 本文采用缓存技术, 这样避免频繁的查询数据库, 相应减小服务器的压力.

每个人都希望自己的请求在服务器端得到最快的响应. 而这又是不能实现的, 所以需要有一个策略来控制 QoS 的等级, 对于不同等级的请求提供区分服务.

2.2 服务等级的划分

在请求内容分类的基础上, 对用户请求进行服务等级划分. 根据不同用户对不同级别 QoS 的要求, 这些由不同客户端发送的 HTTP 请求被划分为不同的服务等级. 考虑到上一节请求的分类和用户权限^[6]. 对于请求都有:

$$F(C(x), P(x)) = w_1 * C(x) + w_2 * P(x) \quad (1)$$

其中 $F(C(x), P(x))$ 为服务等级划分函数, $C(x)$ 为请求分类等级函数, 该函数由 2.1 节当中的请求分类决定. $P(x)$ 为用户权限等级函数, 该函数由用户权限级别确定.

$w_1, w_2 (0 < w_1 < 1, 0 < w_2 < 1)$ 分别为各函数的权重因子, 通常情况下我们会调整 w_1, w_2 在 $F(C(x), P(x))$ 当中的影响权重, 针对不同的服务器集群提供不同的服务器策略. 为不同的请求做出服务等级的划分.

不同请求被黏贴上不同服务等级的标签, 然后进入就绪队列中等待调度.

2.3 动态反馈确定集群节点服务质量

仅仅依靠网络 QoS 并不能保证端到端(例如 WEB)服务的 QoS. 我们需要在服务器端实现 QoS, 对不同服务等级的请求提供不同质量的服务. 如何反应集群里一个节点服务器的负载量, 也就是其可提供的 QoS. 并且能够实时的将负载量反馈给前端负载均衡器. 是本节要解决的重点.

相关研究表明, 在基于 UNIX 的服务器上, 各硬件资源对应用程序的影响比例如图 2.

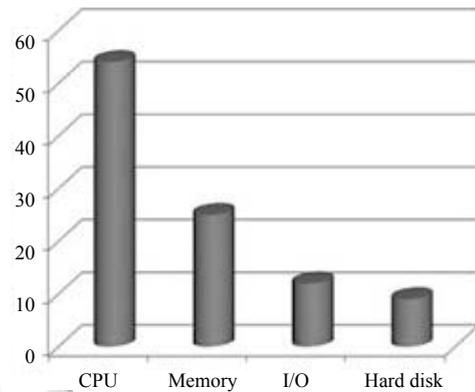


图 2 硬件资源对应用程序的影响比例

根据以上的研究, 在计算服务器端的 QoS 主要考虑计算机的以下参数: CPU 的处理速度、内存容量、系统 I/O 速度和网络带宽. 服务器端 QoS 权值计算公式^[7]:

$$Q(N_i) = R_1 \frac{L_c(N_i)}{B_c} + R_2 \frac{L_m(N_i)}{B_m} + R_3 \frac{L_g(N_i)}{B_g} + R_4 \frac{L_n(N_i)}{B_h}$$

其中 $L_f(N_i)$ 表示节点 N_i 某一参数的当前值, B_f 表示某一参数的基准值, 依次为基准 CPU 处理速度、基准内存容量、基准 I/O 速度、和基准网络接口速度, 这些基准可根据各节点的实际情况统计确定.

R_i 是每个参数设定的可调系数, 可用以区分各种硬件资源的重要程度, 但是必须满足 $\sum R_i = 1$. 一组经典的设置是采用: (0.4、0.2、0.1、0.3).

2.4 请求调度与分配

到此阶段, 前端请求已经分类并且划分了服务等级, 后端服务器集群也能实时反馈到节点的负载信息. 现在开始进行请求的调度和服务器节点的分配.

对于黏贴有服务等级标签的请求, 根据其服务等级的高低, 相应分配优先调度权. 服务等级越高的, 获得越高优先权. 本文采用可变高优先权调度算法, 是指赋予请求的优先权, 是可以随着请求的调度和其等待时间的增加而增加的, 以便获得更好的调度性能. 例如, 在就绪队列中的请求随着其等待时间的增长, 其优先权以速度 a 提高. 如就绪队列当中有三个请求按照进入队列的顺序排列为 a, b, c , 优先级分别为 P_a, P_b, P_c , 若请求具有相同的优先权初值, 即 $P_a = P_b = P_c$. 过了一段时间后, 优先级分别为 $P_a + 3a, P_b + 2a, P_c + a$. 最先进入就绪队列的请求 a 应其动态优先级 $P_a + 3a$ 最高而优

先获得调度权. 若请求具有不同的优先权初值, 对于优先权初值低的请求, 在其等待了一段时间后, 其优先权便会升到最高, 从而获得调度权. 避免优先权初值低的请求, 在就绪队列中长时间得不到调度, 出现队列饥饿现象. 兼顾了该调度策略的公平性.^[8,9]

对于获得了调度权的请求, 要将其分配到集群节点上. 本文通过动态反馈策略, 在周期 t 内, 反馈集群各节点的负载情况, 即可提供不同 QoS 的能力. 选择负载最小的节点即 QoS 最高的节点, 将请求分配到该节点上去. 周期 t 应该适当选取, 如果选的过大, 节点负载信息更新过慢, 就有可能发生将请求分配到当前负载较大的节点上去, 而发生负载倾斜. 如果 t 选的过小, 频繁的反馈节点信息, 就会无端耗费服务器资源.^[10]

2.5 冗余处理

集群当中, 如果某个节点负载过重而挂载, 而这时请求如果分配到这个节点上时, 就会连接超时, 而反馈不了用户请求. 冗余处理就是将处于重负载或发生故障的服务器当前处理的任务转移到备用服务器上继续执行, 及时的对用户的请求进行响应, 保持系统的流畅运行, 而这些对于用户是透明的. 备份服务器上的资源和主服务器上的必须保持同步. 当过载或发生故障的服务器修复之后, 就可以将任务转移回原来的服务器上. 在一个设计良好的集群当中, 冗余处理是非常有必要的. 失效的服务器可以在系统其余部分正常工作时被修复, 并且主服务器和备份服务器可以彼此相互检测^[11].

2.6 冗余处理

应用场景基本上可分为有状态会话和无状态会话场景^[12]. 对于无状态会话, 每一个请求都会被当作对立事件, 所以负载均衡器可以将无状态会话的请求分发到不同节点上去. 而对于有状态会话的请求, 每一次请求都带有状态数据, 如果将一个有状态会话的请求迁移到另一个节点上处理, 则需要同时将状态信息迁移. 本文采用 memcached 缓存将会话 session 放到一个公共的地方, 每次不同的请求查询这个 session, 用户离开时销毁 session. 当一个带有 session 的请求从一个节点迁移到另外一个节点时, 另一个节点会从 memcached 缓存中读取该请求的 session, 从而实现会话保持, 使得用户请求无缝连接.

3 模型的建立与测试

实验时, 选择三台服务器做后端节点, 一台服务

器做前端控制器, 一台服务器配置成为代理服务器, 实体模型如图 3. 各台服务器的配置如表 1.

表 1 集群各个节点配置

服务器编号	CPU	内存	硬盘
控制器	Intel(R) 2.66HZ	8GB	320GB
代理服务器	Intel(R) 2.66HZ	4GB	250GB
节点 1	Intel(R) 2.66HZ	4GB	250GB
节点 2	Intel(R) 2.66HZ	4GB	250GB
节点 3	Intel(R) 2.66HZ	4GB	250GB

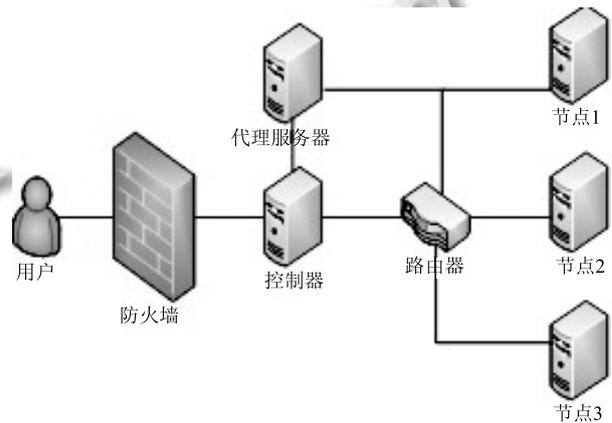


图 3 集群实体模型

实体模型架构配置好之后, 模拟用户请求流对本论文提出的基于区分 Web QoS 的负载均衡集群模型进行测试, 请求流所走流程如下图 4.



图 4 模型流程图

图 4 具体描述如下:

① 当 CL 接收到用户一个新的数据流发送的 R_i 后, 从 TCP 连接请求中解析出请求的类型.

② 我们将请求分类后, 采用 Nginx 作反向代理. 在客户端和内容服务器之间提供了一道安全的屏障. 保护私有数据免受恶意攻击.

③ 而对于事务型数据的请求, 采用 memcached 进行缓存, 缓解服务器的压力.

④ 根据请求类型和用户权限, 将请求划分服务等级.

⑤ 不同请求被黏贴上不同服务等级的标签, 然后进入就绪队列中等待调度.

⑥ 根据可变高优先级调度算法, 在就绪队列当中选择优先级最高的服务请求进行调度.

⑦ 利用上述动态反馈机制, 动态反馈节点信息, 选择服务质量最好节点.

⑧ 如果检测到该节点过载, 就采用上述冗余技术, 将请求转发到备份服务器上. (在节点上另外搭建虚拟机作为备份服务器)

⑨ 如果不过载, 则将请求 R_i 分配到该节点上. 节点服务器处理请求, 返回给用户数据.

测试时, 用 loadrunner 虚拟 250 个用户并发发送请求. 对单个服务器节点, 使用反向代理负载均衡的集群, 和使用本论文提出模型, 就用户通过率、事务平均响应时间、服务器负载情况进行比较.

通过图 5 可以看出, 当虚拟 250 个用户绝对并发对单个服务器节点发送请求时, 有 20 人访问失败, 图 6 当中得到, 新闻模块的访问时间 10.326s, 已经超过了用户可接受的时间, 可以认为 250 人绝对并发访问, 已经是到达单个服务器的瓶颈.

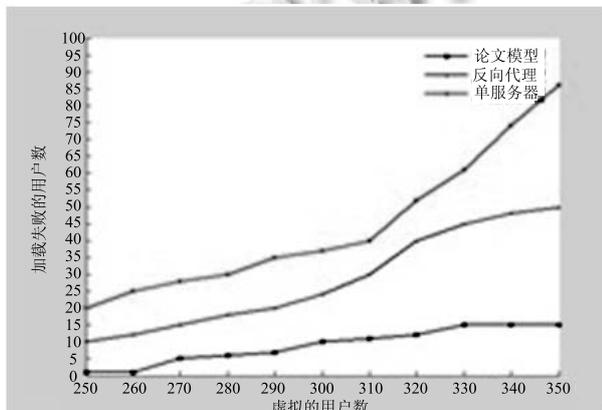


图 5 虚拟用户通过率对比

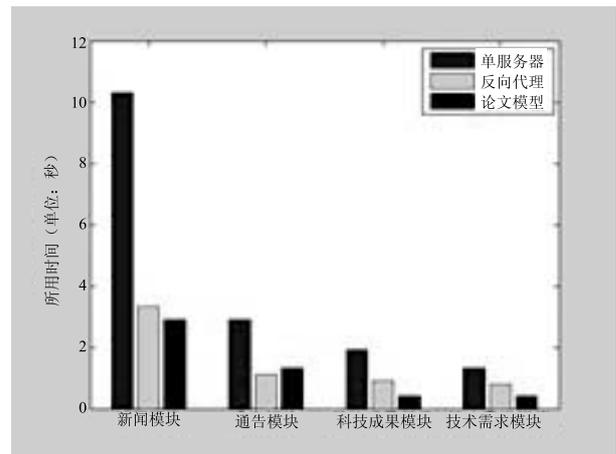


图 6 事务平均响应时间对比

图 5 可以看出, 当使用反向代理负载均衡策略时, 随着虚拟用户的增加, 请求失败的用户数明显减少. 建立本论文集群模型后, 随着虚拟用户的增加, 请求失败的用户数相比较反向代理负载均衡策略明显要少, 而且新闻模块的响应时间是 2.993s, 达到用户心里可以接受范围. 这就表明该集群模型有效地解决了上述单个服务器的瓶颈, 且相比于反向代理负载均衡策略有更短的响应时间, 更高的稳定性. 技术和需求模块, 比新闻和通知模块具有更高用户需求和重要性. 从图 6 可以看出, 对于本文提出的集群模型, 技术和需求模块的响应时间要明显小于新闻和通知模块的响应时间, 这表示对于不同的请求本模型进行了区分服务.

图 5 和图 6 中可以看到, 论文提出的集群模型, 具有较低的加载失败率和较快的相应时间. 从图 7 当中可以看出, 集群模型的各项参数基本都是单个节点各项参数的 1/3, 表明该集群模型达到了负载均衡的目的, 并且比反向代理负载均衡策略各集群节点承担更小的负载. 具有更好的稳定性和实用性.

4 结语

本文提出建立基于区分 Web QoS 的负载均衡集群模型并进行了测试. 建立该模型的根本目的就是为了使得在一定的时间内用户得到的响应尽可能的多, 响应时间尽可能的短. 并且对用户请求提供有区分的服务. 在此意义上, 该模型考虑到了对私有数据采用反向代理策略加以保护, 对事务型数据进行缓存, 根据请求分类和用户权限将划分请求的服务等级, 动态反馈策略确定集群节点可提供的服务质量, 再进行请求

的调度和分配. 以及对服务器节点过载迁移的冗余和对服务器间请求转发时的会话保持. 仿真实验表明, 该模型具有良好的负载均衡能力. 可以保证用户私有数据的安全性, 并对不同用户请求提供区分服务.

参考文献

- 1 郑祺,周广平.基于内容分类的集群负载均衡算法[学位论文].杭州:浙江科技学院.2011.
- 2 刘敏,段富.基于请求分类的 Web QoS 控制策略模型研究[学位论文].太原:太原理工大学.2008.
- 3 郭平,李琪.基于服务器负载状况分类的负载均衡调度算法.华中科技大学学报,2012.
- 4 程久军,李玉宏,程时端,马建.移动 P2P 系统体系机构与关键技术的研究.北京邮电大学学报,2006,4(29):1-2.
- 5 姚琳琳,何倩,王勇,赵帮.基于分布式对等架构的 Web 应用防火墙.计算机工程,2012,4(22):114.
- 6 张健.云计算服务协议(SLA)研究.电信网络技术,2012,2(2):7.
- 7 张玉芳,魏钦磊,赵膺.基于负载权值的负载均衡算法.计算机应用研究,2012,12(4):4711.
- 8 赵涓涓,刘涛,强彦,卢军佐.云计算中基于 Session 和内容等级的数据库请求分类算法[学位论文].太原:太原理工大学.2013.
- 9 Siala F, Ghedira K. A multi-agent selection of Web service providers driven by composite QoS. Computers and Communications (ISCC), 2011 IEEE Symposium on. 2011. 55-60.
- 10 Mehta H, Kanungo P, Chandwani M. Decentralized content aware load balancing algorithm for distributed computing environments. Proc. of the International Conference & Workshop on Emerging Trends in Technology. 2011.370-375
- 11 Deng YH, Lau, Rynson W. On delay adjustment for dynamic load balancing in distributed virtual environments. visualization and computer graphics. IEEE Trans. on Visualization and Computer Graphics, 2012, 18(4): 529-537.
- 12 孙天昊,陈飞,邓俊昆.一种基于 cookie 会话保持的 LVS 集群系统[学位论文].重庆:重庆大学.2013.
- 13 赵艳芳,蒋云起.Web 服务器负载均衡会话保持的研究[学位论文].昆明:云南民族大学.2011.
- 14 高昂,慕德俊,胡延苏,潘文平.Web QoS 中的预测控制与比例延迟保证.计算机科学,2010,1(1):57.
- 15 许少华,夏智伟.基于轮转周期的动态反馈负载均衡算法.计算机科学与技术,2013,1(6):64.

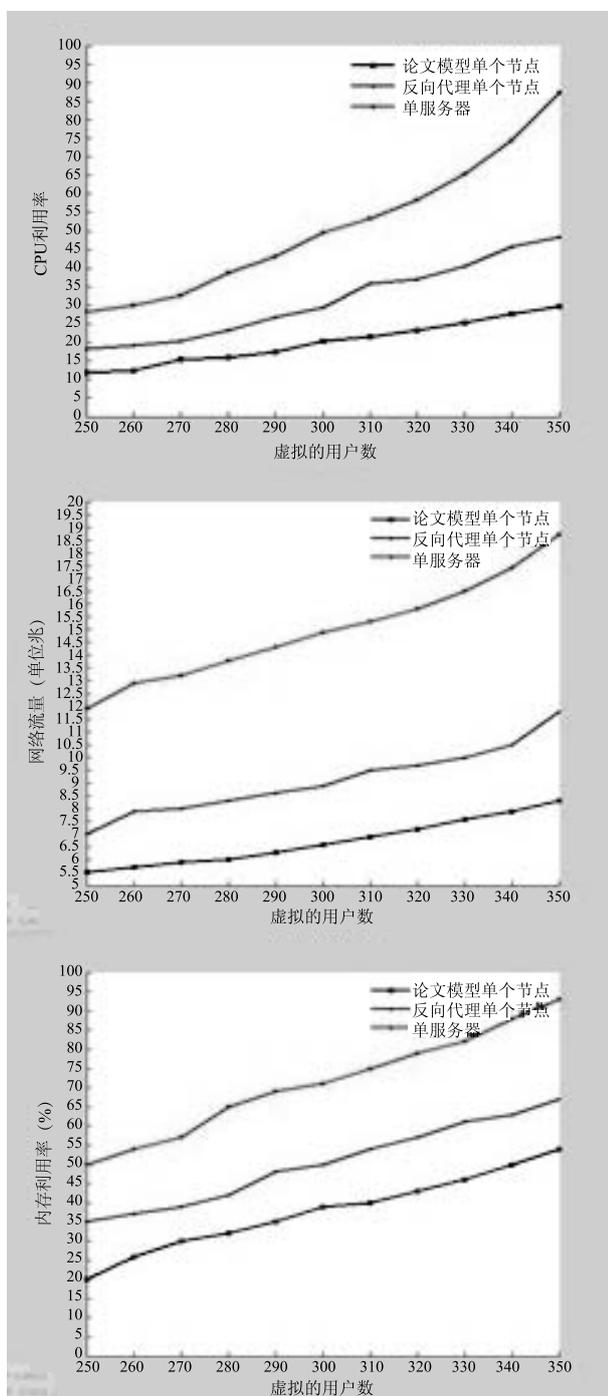


图 7 服务器端负载情况对比