湖南工程學院

毕业设计(论文)开题报告

题　　目：UNIX 下高性能 HTTP 反向代理服务器的设计与实现

学　　院：计算机与通信学院

专　　业：计算机科学与技术

学生姓名：丁　涛　　学　号：201303010202

指导老师：周 铁 山

2017 年　4　月　11　日

开题报告填写要求

1．开题报告（含“文献综述”）作为毕业设计（论文）答辩委员会对学生答辩资格审查的依据材料之一。此报告应在指导教师指导下，由学生在毕业设计（论文）工作前期内完成，经指导教师签署意见及所在专业审查后生效。

2．开题报告内容必须用黑墨水笔工整书写或按此电子文档标准格式（可从教务处网页上下载）打印，禁止打印在其它纸上后剪贴，完成后应及时交给指导教师签署意见。

3．“文献综述”应按论文的格式成文，并直接书写（或打印）在本开题报告第一栏目内,学生写文献综述的参考文献应不少于 10 篇（不包括辞典、手册），其中至少应包括 1 篇外文资料；对于重要的参考文献应附原件复印件，作为附件装订在开题报告的最后。

4．统一用 A4 纸，并装订单独成册，随《毕业设计（论文）说明书》等资料装入文件袋中。

毕 业 设 计（论 文）开 题 报 告

1．文献综述：结合毕业设计（论文）课题情况，根据所查阅的文献资料，每人撰写 2500 字以上的文献综述，文后应列出所查阅的文献资料。

一、 前言

HTTP 协议作为互联网最流行的应用层协议，已占据互联网流量转发的主要数据包。单机服务器在收到大量 HTTP 连接请求，势必会导致单机 Web 响应不及时；服务器出口带宽限制，大量主动丢包也会使响应时间加长；主动丢包还会影响到路由链路中间转发路由器及交换机计算压力增大，进一步影响 Web 请求的响应时间。反向代理服务器也会收到大量客户端发送的请求，因此加快高并发环境下响应请求也至关重要。本文就分布式 HTTP 服务器的整体框架做简要描述，着重分析反向代理服务器的分布式框架。随后分析适合高并发环境的网络异步非阻塞式 IO 框架，基于此框架设计反向代理服务器，并进行性能测试，以验证本课题的可行性及有效性。

二、 HTTP

超文本传输协议（HTTP）具有分布式，协作式等特性，是一种适合超媒体信息系统数据传输的应用层协议。

HTTP 协议最早的版本是 HTTP/0.9，作为试验性协议，HTTP/0.9 并没有制定标准文档。直到 1996 年 5 月 IETF 批准通过了 HTTP/1.0 标准，这是 HTTP 协议的第一份标准协议文档。在该标准发布不久后，随着 HTTP 协议被广泛使用，其缺点也慢慢体现出来。当客户端与服务器端完成本次数据传输后，客户端会主动断开 TCP 连接。在完成第一次 HTTP 数据传输请求后，客户端很有可能还会访问同一个网站的不同页面，断开 TCP 连接会导致后续的 HTTP 访问请求又要经过 TCP 三次握手建立连接。因此 IETF 定制了第二份 HTTP 协议版本 HTTP/1.1，新增了 HTTP 层面的 TCP 长连接的保活机制。HTTP/1.1 是目前互联网上最流行的 HTTP 版本。

HTTP 在设计上允许中间缓存服务器改善客户端到服务器之间的通信质量。 高流量网站通常都会受益于分布式 Web 缓存服务器上游服务器带来的负载均衡功能，以提高响应时间。 Web 浏览器在访问源服务器之先访问 WEB 缓存服务器的 Web 资源，并在可能的情况下重用它们以减少网络流量。私有网络内部的 HTTP 代理服务器可以通过使用外部服务器作为代理服务器通过中继 HTTP 消息来达到与客户端通信的目的，

而不必需要多个公网可路由的地址。

三、 IO 模型

1) 阻塞和非阻塞

阻塞 IO 和非阻塞 IO 关注的是当前进程（或线程）在等待 IO 调用结果（或消息）时的状态。阻塞 IO 是指调用者发起 IO 请求时，在得到 IO 结果之前当前进程（线程）将被挂，进入非可执行态，直到被调用函数返回 IO 结果才能恢复运行进入就绪态或可运行态。非阻塞 IO 是指调用者发起 IO 请求时，被调用函数立即返回，有可能是返回的 IO 结果，也有可能是返回的错误代码，此时调用者可以选择过一定时间再次发起 IO 请求，直到获取到预期的 IO 结果；在这段时间里调用者可以执行其他的函数，而不会被系统挂起。

2) 同步和异步

同步和异步关注的是函数（进程或线程）之间的消息传递机制。同步调用，就是在调用者发出一个调用请求时，在没获取到结果之前，该调用就不会返回。一旦调用返回，那么就获取到了调用返回结果。因此，调用者在主动等待当前调用结果。异步调用，与同步调用相反，在调用发出后，直接返回，但是没有返回结果。因此，当一个异步调用过程发出后，调用者不会马上得到调用结果，而是在调用发出后，由被调用者通过发送状态、信号、通知等告知调用者，或通过执行注册的回调函数（钩子函数）来处理当前返回值。

四、 Libevent

不同操作系统对异步非阻塞的支持与 API 接口形式都不同，例如 POSIX 的 select，Windows 的 IOCP，Linux 的 epoll 等。在编写有跨平台需求的高并发的程序时，通常都需要对不同平台的 IO 接口函数进行适配，以达到软件跨平台的目的。

Libevent 作为一款开源的网络库，封装了不同平台底层的异步函数接口，向上提供了接口简洁以注册回调函数为基础的事件 API，是一个非常适合具有跨平台需求的程序开发的网络库。Libevent 本身使用 C 语言编写，效率非常高，在设计上支持事件可扩展，专注网络 IO 的优化，支持多线程访问。在跨平台上，通过条件编译，去支持 Windows，Unix-like 等操作系统。

bufferevent 是 libevent 库中具有缓存功能的事件结构体，应用层通过创建 bufferevent 结构体并注册相应的事件回调函数来通知 libevent 当读写事件发生时，应该调用的函数地址。libevent 通过检查 bufferevent 中的读写缓存的大小来判断是否应该调用上层注册的回调函数，从而将 IO 读写转换成异步非阻塞的 IO 模型。

五、 小结

大量 HTTP 数据包在互联网上转发，对互联网线路上的路由器及交换机的压力是比较大的，在物理上无法快速扩大链路带宽的情况下，可将用户请求及响应转发至距离用户最近的服务器群组，降低中间链路数据包转发压力。将具有缓存功能的反向代理服务器布置在离用户较近的地方，代替源服务器向客户端发送响应数据，能降低源服务器负载，加快用户收到响应的时间，同时可大大降低客户端到源服务器的链路转发压力。Libevent 异步网络事件库对大量并发的支持比较好，其具有缓存的事件结构体 bufferevent 设计巧妙简洁，非常适合用于 HTTP 等基于数据流的网络协议的底层库。

六、 参考文献

[1]. Xie Xiren. *Computer Network[M]*. Sixth Edition. Beijing, Electronic Industry Press. June 2013.

[2]. Gourley, D. *et al*. *HTTP:The Definitive Guide[M]*. Sebastopol, O'Reilly Media, Inc. September 2002.

[3]. T. Berners-Lee, R. Fielding and H. Frystyk, *Hypertext Transfer Protocol -- HTTP/1.0*, IETF RFC 1945, May 1996; www.rfc-editor.org/rfc/rfc1945.txt.

[4]. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, *Hypertext Transfer Protocol -- HTTP/1.1*, IETF RFC 2068, January 1997; www.rfc-editor.org/rfc/rfc2068.txt.

[5]. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach and T. Berners-Lee, *Hypertext Transfer Protocol -- HTTP/1.1*, IETF RFC 2616, June 1999; www.rfc-editor.org/rfc/rfc2616.txt.

[6]. R. Fielding, Ed. and J. Reschke, Ed., *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*, IETF RFC 7230, June 2014; www.rfc-editor.org/rfc/rfc7230.txt.

[7]. R. Fielding, Ed. and J. Reschke, Ed., *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*, IETF RFC 7231, June 2014; www.rfc-editor.org/rfc/rfc7231.txt.

[8]. R. Fielding, Ed. and J. Reschke, Ed., *Hypertext Transfer Protocol (HTTP/1.1):*

Conditional Requests, IETF RFC 7232, June 2014; www.rfc-editor.org/rfc/rfc7232.txt.

[9]. R. Fielding, Ed., Y. Lafon, Ed. and J. Reschke, Ed., *Hypertext Transfer Protocol (HTTP/1.1): Range Requests*, IETF RFC 7233, June 2014; www.rfc-editor.org/rfc/rfc7233.txt.

[10]. R. Fielding, Ed., M. Nottingham, Ed. and J. Reschke, Ed., *Hypertext Transfer Protocol (HTTP/1.1): Caching*, IETF RFC 7234, June 2014; www.rfc-editor.org/rfc/rfc7234.txt.

[11]. R. Fielding, Ed. and J. Reschke, Ed., *Hypertext Transfer Protocol (HTTP/1.1): Authentication*, IETF RFC 7235, June 2014; www.rfc-editor.org/rfc/rfc7235.txt.

[12]. M. Belshe, R. Peon and M. Thomson, Ed., *Hypertext Transfer Protocol Version 2 (HTTP/2)*, IETF RFC 7240, May 2015; ww.rfc-editor.org/rfc/rfc7240.txt.

[13]. R. Peon and H. Ruellan, *HPACK: Header Compression for HTTP/2*, IETF RFC 7241, May 2015; ww.rfc-editor.org/rfc/rfc7241.txt.

[14]. W. Richard Stevens. *TCP/IP Illustrated, Volume 1: The Protocols[M]*. New Jersey. Addison-Wesley. 1994.

[15]. W. Richard Stevens and Stephen A. Rago. *Advanced Programming in the UNIX Environment[M]*. Third Edition. New Jersey. Pearson Education, Inc. May 2013.

[16]. W. Richard Stevens, Bill Fenner and Andrew M. Rudoff. *UNIX Network Programming Volume 1: The Sockets Networking API[M]*. Third Edition. New Jersey. Pearson Education, Inc. November 2003.

Nick Mathewson, *Fast portable non-blocking network programming with Libevent*, Libevent.org, Jan 2012; www.wangafu.net/~nickm/libevent-book/TOC.html.

毕 业 设 计（论 文）开 题 报 告

２．开题报告：一、课题的目的与意义；二、课题发展现状和前景展望；三、课题主要内容和要求；四、研究方法、步骤和措施

开 题 报 告

一、 课题的研究意义

客户端向地理位置上较远的服务器发起请求，中间经过多级交换机及路由器的转发，占用中间经过的网络带宽及设备的计算能力。在地理位置上距离客户端较近的地方布置几台具有缓存功能的反向代理服务器，将会大大降低中间网络设备的计算压力，减少客户端到服务器之间的延时。在大量用户同时访问资源时，布置反向代理服务器能将请求响应的压力分散到各个具有缓存功能的反向服务器上，能加快源服务器对客户端请求的计算速度。

二、 课题的研究目的

nginx 反向代理服务器的设计目的主要是为了减轻源服务器的计算压力，以负载均衡的形式均匀代理客户端发送的请求并一一响应。但其在响应内容缓存的设计上就不尽人意了。本课题设计目标着重强调中间结果缓存，降低中间网络带宽占用，降低客户端收到响应的延时，以此提高用户体验。

三、 国内外发展现状

国外着重设计 HTTP 中间缓存的软件比较知名的是 squid。squid 是一个具有缓存功能的 Web 代理服务器，支持 HTTP，HTTPS，FTP 等常见协议代理。它通过响应缓存最常被访问的页面来降低客户端到服务器之间的带宽占用及延时。其配置较为复杂，设计上追求通用，不适用于复杂的生产环境。

国内有两款商用软件针对国内设计开发的缓存服务器，秒开缓存及缓存大师。秒开缓存和缓存大师作为闭源商业软件，其底层实现原理及逻辑没有相应的论文或专著对其进行描述。这两款软件的面向群体为 IPS，网吧等，用户正向代理缓存。它们的缓存代理原理均为由交换机端口镜像功能复制网关出口数据包到服务器，缓存服务器通过检测客户端发送的请求，伪装成源服务器发送 HTTP 302 重定向，重定向至本机，强制使客户端访问缓存服务器，以达到节省带宽的目的。

四、 课题研究主要内容

本课题在实现上采用 libevent 库作为网络库，使用 evconnlistener 作为监听器，监听 80 端口用户发送来的 HTTP 连接请求，同时用 bufferevent 以非阻塞方式缓存用户的请求。

课题设计分两个模块，代理模块，缓存模块。代理模块的主要功能为接收客户端发起的连接请求，并向源服务器转发客户端的请求将源服务器的响应回送给客户端。缓存模块在代理模块收到源服务器相应时，可根据 HTTP 头部信息来确定当前相应是否应该缓存到本地，在用户下次访问同一内容时可直接从缓存中读取相应发送给客户端，以减轻源服务器压力并降低客户端延时时间。

在程序编码完成后，本文将以 nginx 作为比较对象，自行设计通用压力测试脚本分别测试本程序及 nginx 程序在文本文件、图片文件的传输时的反向代理成功率和响应时间，以此来判别本程序与 nginx 的性能差异。

五、 进度计划

2017 年 2 月 14 日-2017 年 3 月 10 日 明确课题任务及要求，搜集课题所需资料，了解本课题研究现状、存在问题及研究的实际意义；写好选题报告和文献综述。

2017 年 3 月 11 日-2017 年 3 月 28 日 毕业实习。

2017 年 3 月 29 日-2017 年 4 月 10 日 查阅相关资料，自学相关内容，确定软件总体设计方案。

2017 年 4 月 11 日-2017 年 4 月 20 日 各部分模块设计及部分调试，系统总体联调。

2017 年 4 月 21 日-2017 年 5 月 21 日 整理资料，撰写毕业设计论文。

2017 年 5 月 21 日-2017 年 6 月 6 日 毕业论文审定、打印，答辩准备。

2017 年 6 月 7 日-2017 年 6 月 8 日 答辩。

毕 业 设 计（论 文）开 题 报 告

指导教师意见：

1．对“文献综述”的评语：

2．对本课题的深度、广度及工作量的意见和对设计（论文）结果的预测：

指导教师：_____

年　月　日

所在专业审查意见：

负责人：_____

年　月　日