构建基于 IPv6下的高性能 FTP系统

宋东兴[1,2], 龚 俭[2]

(1.常熟理工学院 计算机系,江苏 常熟 215500;
2.东南大学 计算机工程系,江苏 南京 210096)

[摘要] Ipv6作为下一代 IP网络发展标准它取代 Ipv4已经成为必然趋势,随着 Ipv4向 Ipv6的平滑过渡,现有的各种应用程序势必最终要能适应下一代网络,构建基于 Ipv6下的应用程序就有着现实的意义.介绍了 Ipv4下的文件传输软件 bbftp的性能、IPv6套接口的结构,分析了 IPv6套接口与 IPv4套接口的主要区别,阐明了把应用程序从 Ipv4移植到 Ipv6存在的问题、实现思路及需要采取的方法.根据实际需求,并结合 IPv6 socket编程,通过对 bbftp的改造实现了一个基于 IPv6的高性能的 FTP系统,经过对移植后的系统测试数据分析表明系统稳定可靠,达到了预期的目的.

[关键词] IPv6套接口, bbftp,移植,高性能 FTP
[中图分类号]TP393 [文献标识码]A, [文章编号]1672-1292-(2005)01-0057-04

Construct IPv6-Enabled High Perfomance FTP System

SONG Dongxing[1,2], GONG Jian[2]

(1. Department of Computer, Institute of Changshu Technology, Jiangsu Changshu 215500, China;
2. Department of Computer Engineering, Southeast University, Jiangsu Nanjing 210096, China)

Abstract As the development standard of next generation IP network, Ipv6 is bound to substitutes for Ipv4. With the smooth transition from IPv4 to IPv6, various kinds of existing application program will certainly accommodate to the next generation network, thus constructing the application procedure based on Ipv6 is of great real significance. This article introduces performance of bbftp, a file transfer software under the Ipv4, and structure of the IPv6 socket interface, analyzes the main difference in socket interface structure between IPv6 and IPv4, and clarifies the difficulties in and problems on transplanting the application program from Ipv4 to Ipv6, the realizing scheme and the method to be adopted. According to the actual demand, in combination with IPv6 socket programming, a IPv6-enabled high performance FTP system is realized by transforming bbftp. The analysis of the tested data for the transplanted system shows that the system is stable and reliable, and reaches the expected ends.

Key words IPv6 Socket Interface; bbftp; transplant; high performance FTP system

0 引言

目前, IPv6作为下一代 IP网络发展标准它取代 IPv4已经成为不可逆转的趋势.随着 IPv4向 IPv6的平滑过渡,现有的各种应用程序势必最终要能适应下一代网络,构建基于 IPv6下的应用程序就有着现实的意义.

现在大多数的 FTP系统,如 CuteFtp, FlashFXP都只是支持 IPv4的,我们选择 FTP应用为研究点,构建基于 IPv6下的高性能的 FTP系统.一种思路是完全重头编写;另一种思路是充分利用现有资源,把基于 IPv4下的高性能 FTP系统移植到 IPv6上.我们选择后者,选择 RCF支持的最新版软件 bbftp 3.0进行改造.

bbftp是 IPv4下的高性能的文件传输软件.它由法国里昂 IN2P3计算机中心的 Gilles Farrache开发.它的诞生,使得用户在欧洲粒子物理研究所(CERN)的数据中心与其它场所之间的传输数据变得更加安全,更加快速.特别对大文件的传输比普通 FTP具有更高的效率.此外 bbftp是开源的.

收稿日期: 2004-05-30
作者简介: 宋东兴(1973-),讲师,主要从事计算机网络等方面的教学与研究. E-mail: sdxing@cslg.cn
通讯联系人: 龚俭(1957-),教授,博士生导师,主要从事计算机网络安全等方面的教学与研究. E-mail: jgong@njnet.edu.cn

为此,我们在自建的IPv6局域网环境下,以"基本的支持IPv6的socket接口扩展"[1]为指导,对开源软件bbftp最新版3.0进行改造,修改相关的代码,实现bbftp从IPv4到IPv6的移植.

1 bbftp

1.1 系统分析

bbftp是一个文件传输软件,它是高性能文件传输协议的一个实现,RFC-1323是该协议的正式规范[2]. bbftp与普通FTP在性能和实现方式上都有着较大的差别. 首先,普通Ftp不允许手工动态调整TCP窗口的大小,这在高带宽或长距离延时的路径上,数据吞吐量受到很大的影响;而bbftp允许在运行时调整窗口大小,最大理论值达1G. 其次,普通Ftp采用一个数据流进行数据传输,即一个文件作为一整块进行顺序传输,这样在传输一个大文件时,它的速度就会受到影响;而bbftp可以很方便地采用多个TCP流在客户和服务器之间传送文件,对于大文件,它把文件分成多块,每一块在一条数据连接上传输,双方利用多数据连接并行传送文件的多个分块,这在负载的广域网上要比单流Ftp具有更高的性能,同样bbftp可以在运行时对数据流个数这个参数进行动态调整. 第三,安全性. 普通Ftp的一个主要问题是用户名和口令在网上明文传送,而bbftp可提供加密用户名和口令、ssh验证、证书认证等多种模式,避免了未授权用户对文件非法访问. 第四,普通Ftp一次提交一个命令,而bbftp一次可提交多条命令,且一条命令执行出错,bbftp能自动出错重试,并且一条命令出错不影响下面命令的继续执行. 此外同普通ftp相比,bbftp还有RFIO、定制超时时间等特有的功能.

因此我们选择bbftp软件,在它的基础上,修改相关代码,增加IPv6特性,把bbftp从IPv4移植到IPv6上,构造基于IPv6的高性能的FTP系统. 高性能的FTP系统对在网上分析和共享大量数据,并且能快速存取和传输海量数据的多人协作方式很有意义.

1.2 移植存在的问题

bbftp只支持IPv4,客户和服务器间传送的是IPv4格式的报文. 而在纯IPv6的环境下,IPv6协议栈不能处理IPv4报文,只能处理IPv6格式的报文. 要把bbftp系统从IPv4移植至纯IPv6上,关键在于:IPv4报文和IPv6报文格式不一样,IPv6协议栈不能识别和处理IPv4报文,为此我们要在bbftp客户和服务器端都要创建IPv6套接口,以此来代替IPv4套接口,把IPv4的通信端点的标识信息用IPv6的信息来替代,变换相关函数中与地址族有关的参数,根据IPv6的特性对应用程序进行适当的调整.

2 迁移设计

2.1 Socket API对IPv6的支持

2.1.1 IPv4和IPv6套接口地址结构比较

套接口地址结构中存放着通信端点的标识等重要的信息. 在这个结构中,一般包含:该结构的大小;所采用的地址族(协议族);相应端的地址信息等. 对TCP/IP来说,端地址指的是IP地址和传输层端口号的组合.

在<sys/socket.h>中定义了IPv4地址族AF_INET和IPv6地址族AF_INET6,随着地址族的不同,套接口地址结构的具体内容也不一样,如AF_INET使用的结构是sockaddr_in,AF_INET6使用的结构sockaddr_in6.

IPv4套接口地址在头文件〈netinet/in.h〉中定义如下:

```
struct sockaddr_in{
    uint8_t         sin_len;        /结构体总长
    sa_family_t     sin_family;     /地址族,此处为AF_INET
    in_port_t       sin_port;       /16位传输层端口号
    struct in_addr  sin_addr;       /IPv4地址
    char            sin_zero[8];    /填充0
};
struct in_addr{
    in_addr_t       s_addr;         /32位IPv4地址
};
```

IPv6套接口地址在头文件〈netinet/in.h〉中定义如下:

```
#define SIN6_LEN
struct sockaddr_in6{
    uint8_t         sin6_len;       /结构体总长
    sa_family_t     sin6_family;    /地址族,此处为AF_INET
    in_port_t       sin6_port;      /16位传输层端口号
    uint32_t        sin6_flowinfo;  /优先权和流标号
    struct in6_addr sin6_addr;      /IPv6地址
};
struct in6_addr{
    uint8_t s6_addr[16];        /128位IPv6地址
};
```

由以上定义可知,除结构体长度和IP地址长度的不同外,同IPv4相比,IPv6增加了sin6_flowinfo域,该域有3个字段:低24位是流量标号,下4位是优先级,再下4位保留[3].

2.1.2 IPv6通配地址

调用bind()函数给套接口分配一个本地协议

地址, 协议地址是 IP地址与 TCP或 UDP端口号的组合, 我们使用通配地址通知内核选择一个本地 IP地址, 在 IPv4中, 通配地址由常量 INADDR_ANY来指定, 其值一般为零; 若我们要绑定一个本地 IPv6的协议地址, 必须使用 IPv6的通配地址 in6addr_any, 该变量在头文件 〈netinet/in.h〉中含有 extern声明. 它由系统分配并初始化为常量 IN6ADDR_ANY_INIT.

2.1.3 IPv4移植到 IPv6涉及到的库函数

由于 IPv4与 IPv6的地址大小不同, 套接口结构也有差异, 在从 IPv4到 IPv6迁移过程中就会涉及到套接口创建、名字到地址的转换、不同地址间的转换等问题, 为此系统在函数库中增加了新的函数或扩展现有函数来实现这些功能.

(1) 创建套接口函数

函数 socket()返回一个类似于文件描述字的句柄, 来标识所创建的套接口, 其原型:

int socket(int family, int type, int protocol);

该函数的返回值为一个整数, 即套接口描述字, 通过这个描述字来访问相应的套接口. 参数 family指定本次通信所使用的地址族 (或协议族). 对 IPv4来说, 它的值为 AF_INET, 而对 IPv6来说, 则是 AF_INET6.

(2) 名字到地址的转换函数

实际中人们习惯用域名而不是数值形式的地址来区分不同的主机, 而 bind()函数处理的只能是数值形式的地址, 解决这个问题的方法是使用名字和地址的转换函数, 查找主机名的基本函数是 gethostbyname()和 gethostbyname2(). 对于前者它既能查询 IPv4地址, 也能查询 IPv6的地址, 但它不能让调用者指定特定的地址类型, 而后者可以通过一个标识参数让调用者指定需查询的特定类型的地址. 故从 IPv4迁移到 IPv6过程中 gethostbyname2()更合适一些. 两者的缺点是在线程中不够安全[3]. 我们可采用 getipnodebyname()函数来调用解析器代码, 向 DNS服务器执行一个对 A记录或是 AAAA记录的查询, 它返回的是 IPv4或是 IPv6的地址, 并保证了线程的安全.

(3) IP地址表达格式与地址数值格式转换函数

两组地址格式转换函数 inet_pton()和 inet_ntop(), 它们在 ASCII字符串与网络字节序的二进制值间转换地址, 对 IPv4和 IPv6地址都能处理. 可以用它们来替换只能处理 IPv4的函数 inet_aton()、inet_addr()、inet_ntoa().

2.2 实现思路

要构建基于 IPv6下的高性能 FTP系统, 基本思路是: 保留 bbftp的基本算法不变, 维持其多流数据传输、用户名和口令的加密传送, 增大的 TCP窗口等高性能文件传输特性, 以 RFC 2553"基本的支持 IPv6的 socket接口扩展"为指导对其进行改造, 去掉 bbftp中与 IPv4依赖的代码, 增加 IPv6有关的特性, 把 bbftp从 IPv4移植到 IPv6上.

2.2.1 在 RedHat Linux9.0系统下安装 IPv6协议栈

在现行的 Linux发行版, IPv6功能被编译成一个可载入模块, 它不会自动载入. 要测试系统是否支持 IPv6可用 test命令, 格式: # test -f /proc/net/if_inet6 && echo "Running Kernel is IPv6", 如失败表明模块没有载入. 要载入 IPv6模块可用 modprobe命令, 格式: #modprobe IPv6. 如安装成功, 在其上运行的服务器就能处理 IPv6客户请求, 接收并处理 IPv6报文.

2.2.2 bbftp从 IPv4移植到 IPv6

修改的步骤如下:

(1) 构造一双协议栈的服务器, 给其分别配置 IPv4和 IPv6地址.

(2) 修改原来创建的 IPv4套接口, 创建基于 IPv6的套接口.

(3) 把套接口函数中, 涉及 IPv4套接口地址结构的指针的参数, 用 IPv6套接口地址结构的指针来替换.

(4) 将 IPv6套接口绑定到 IPv6通配地址和 5021监听端口 (注: 5021是 bbftp用于控制连接的端口).

(5) 采用兼容 IPv4/IPv6的地址转换函数, 代替只支持 IPv4的地址转换函数.

(6) 对有关 IPv4地址显示格式代码进行修改, 以足够的空间和长度显示 IPv6格式地址.

3 系统测试

经过上面的修改工作之后, 为了验证移植是否成功, 我们在自建的 IPv6局域网上进行测试:

3.1 试验环境

(1) 4台 PC机组建的以太网, 千兆网卡入网.

(2) 其中两台计算机上安装了 RedHat Linux 9.0操作系统, 安装好 IPv6协议栈, 并分别分配了全局单播的 IPv6地址 3ffe:ffff:0:f101::fa和 3ffe:ffff:0:f101::fb.

3.2 测试步骤

(1) 在地址为 3ffe:ffff:0:f101::fa的机器上安

装修改后的 bbftp服务器,把修改后的 bbftp客户端安装在地址为 3ffe: ffff: 0: f101:: fb的机器上.(安装过程参考 http://doc.in2p3.fr/bbftp/download.html).

(2) 启动 bbftp服务器在后台运行#./bbftpd start

(3) 用 IPv6地址进行测试:

(a) 以下载服务器 sdxing用户当前目录下大小为 99522560(97.2M)字节的文件 font.tar,到客户端当前用户目录 /home/wj下,指定下载参数为一个流(-p 1)为例,命令如下:

#./bbftp -e 'get font.tar /home/wj/' -w -p 1 -u sdxing 3ffe: ffff: 0: f101:: fa

正确运行后出现反馈信息:

get 99522560 1.279 256 256 256 1 nogzip

含义是: get 99522560个字节大小文件花费 1.279 s,接收缓冲区大小、TCP发送窗口和 TCP接收窗口都是 256 k字节, 1个流下载无压缩.这表明 IPv6下 get命令测试通过.

(b) 上传 /home/wj的文件 font.tar到服务器的当前用户目录下命令如下:

#./bbftp -e 'put /home/wj/font.tar /home/sdxing/' -w -p 1 -u sdxing 3ffe: ffff: 0: f101:: fa

正确运行后出现反馈信息:

put 99522560 1.659 256 256 256 1 nogzip

这表明 IPv6下 put命令测试通过.

(4) 用域名进行测试:

在 DNS服务器中增加一条域名 aaa6.mytest.com到 IPv6地址 3ffe: ffff: 0: f101:: fa解析的 AAAA记录,并启动 DNS服务,输入如下命令:

#./bbftp -e 'get font.tar /home/wj/' -w -p 1 -u sdxing aaa6.mytest.com

测试成功.

3.3 bbftp测试数据

在 3.2所述试验环境下,我们在缓冲区大小、TCP发送窗口和 TCP接收窗口大小都为 256 kb字节情况下,以 get和 put一个 99522560(97.2M)字节大小文件,不断调整流数目,测试得到了如表 1所示的一组数据.

表 1 移植后的 bbftp性能

次数	下载时间/s	流数目	传输速率/(Mb/s)	上传时间/s	流数目	传输速率/(Mb/s)
1	1.279	1	622.5	1.659	1	479.9
2	1.469	2	541.9	1.488	2	535.1
3	1.350	3	589.8	1.186	3	671.3
4	1.088	4	731.8	1.315	4	605.5
5	1.088	5	731.8	1.273	5	625.4
6	1.378	6	577.8	1.312	6	606.8
7	1.447	7	550.2	1.221	7	652.1
8	1.407	8	565.9	1.179	8	675.3
9	2.976	9	267.5	1.257	9	633.4
10	1.406	10	566.3	1.417	10	561.9
平均			574.6			604.7

从上面的测试数据来看,在千兆带宽的局域网试验环境下, get和 put传输的平均效率为 57.5%和 60.5%,上传和下载是不对称的. get增加流到 4,5时速度最快,再增加流时速度不稳定,高低有反复, put也有类似的情况,从平均效率上来看,移植后的 bbftp达到较好的性能,但是还有进一步提高性能的空间.

4 结束语

将 IPv4下的应用程序移植到 IPv6上,从原理上可看出,除对 IPv4依赖性很强的程序,如多播、IP选项和原始套接口的程序需花费更多的转换工作外(对 bbftp的移植由于不涉及到多播、IP选项和原始套接口等内容的转换,因为篇幅的关系故对这些内容不展开讨论),大部分 IPv4应用程序转换成能用于 IPv6并不困难,我们选择将 IPv4下 bbftp系统移植到 IPv6下,对构建基于 IPv6下的高性能 FTP系统做了一个有益的尝试.

[参考文献]

[1] Robert E. Gilligan, Suan Thomson, Jim Bound, *et al*. "Basic socket interface extensions for IPv6", RFC 2553[S], March 1999.

[2] Jacobson V, Braden R, Borman D. TCP extensions for high performance. RFC 1323[S]. May 1992.

[3] Richard Stevens W. UNIX网络编程(第1卷)[M]. 第2版. 北京: 清华大学出版社, 1999. 47-51.

[责任编辑: 刘健]