

# linuxsocket端口:基于Linux的Socket网络编程的性能优化

疯狂代码 <http://CrazyCoder.cn/> <http://CrazyCoder.cn/NetworkProgramming/Article78116.html>

来源:维库电子

## 1 引言

随着Internet日益发展和普及网络在嵌入式系统中应用非常广泛越来越多嵌入式设备采用Linux操作系统Linux是个源代码公开免费操作系统具有强移植性所以对基于LinuxSocket网络编程研究越来越重要

Socket实际是网络传输层供给应用层编程接口传输层则在网络层基础上提供进程到进程逻辑通道而应用层进程则利用传输层向另台主机某进程通信Socket就是应用层和传输层的间桥梁如图2所示

使用Socket编程时可以开发客户机和服务器应用它们可以在本地网络上进行通信也可以通过Internet在全球范围内进行通信编写并运行 Socket客户端和服务端双方通过套接字建立了服务连接请求并且通过这些思路方法提高Socket性能

## 3 Socket编程

### 3.1 Socket类型

常见Socket有3种类型:

(1)流式Socket(SOCK\_STREAM)它提供可靠通信流使用面向连接TCP协议从而保证数据传输正确性和顺序性:

(2)数据报Socket(SOCK\_DGRAM)数据通过相互独立报文进行传输是无序并且不保证可靠无差错它定义种面向无连接服务使用数据报协议UDP ;

(3)原始Socket(SOCK\_RAW)它允许直接访问底层协议功能强大但使用较为不便主要用于些协议开发本编写Socket属于流式Socket

### 3.2 Socket编程流程

Socket编程采用客户 / 服务器模式因此编程分为服务器端和客户端两部分

每个Socket都用个半相关描述(协议本地地址本地端口)来表示Socket也有个类似于文件打开该返回个整型Socket描述符随后建立连接数据传输等操作都通过Socket来实现

编程流程如下:服务器端首先建立Socket返回该Socket描述符:配置Socket端口和IP地址;建立\*函数检测是否有客户端向服务器发送请求若有则接收该请求将其放到接收队列中:从接收队列中接受个请求;并向客户端发送确认连接信息

客户端建立个Socket返回该Socket描述符;配置Socket端口和IP地址;向服务器发送连接请求并接收服务器发回确认连接信息双方通信结束后关闭其Socket进行Socket编程基本有bindlistenaccept connectsendrecvclose图3为Socket编程流程图

### 3.3 编译和运行结果

(1)在LinuxVI编辑器下编写服务器端serv.c和客户端clt.c运用交叉编译工具arm-linux-gcc,执行编译指令生成可执行文件

其指令为:

```
#gcc serv.c -o serv
```

```
#gcc clt.c -o clt
```

编译没有则会生成可执行文件serv和clt

(2)配置服务器和客户端IP保证网络畅通在serv.c中已将服务器IP设置为:192.168.2.111在客户端“网络设置”中设置IP为:192.168.2.22可以通过ping命令检测网络是否畅通

(3)在台计算机终端先运行服务器(. / serv)再在客户端计算机终端上运行客户端(. / clt 192.168.2.111)就会看到结果(HelloWang Lei!You are connected!);运行结果如图4和图5所示如果未运行服务器而先运行客户端将立即提示“Connect:Connection refused”

## 4 SOCKET性能优化

### 4.1 解决多路复用

上面运行过程仅实现了个客户端接人在实际情况中人们往往遇到多个客户端连接服务器端情况由于connectrecvsend 都是阻塞性若资源没有准备好则该函数进程将进入睡眠状态无法处理I / O多路复用服务器端serv . c中加入select它可同时\*多个套接字实现I / O多路复用

其原型如下:

该监视系列文件描述符特别是readfds、writefds和exceptfds如果想知道是否能从标准输入和套接字描述符sockfd读入数据只要将文件描述符“0”和“sockfd”加入集合readfds中参数numfds应等于最高文件描述符值加1设置该值为sockfd+ 1它定大于标准输入文件描述符“0”当select返回时readfds值修改为反映选择哪个文件描述符可读重新编译和运行客户端后服务器端允许多个客户端接入服务器端运行结果如图6所示

### 4.2 最小化报文传输延时

通过TCP 进行通信时数据都被拆分成数据块这样它们就可以封装到给定连接TCP payload(指TCP数据包中有效负荷)中TCP payload大小取决于几个原因(如最大报文长度和路径)为了达到较好性能应使用尽可能多可用数据来填充每个报文当没有足够数据来填充 payload时(也称为最大报文段长度maximum segment size或MSS)TCP将采用Nagle算法自动将些小缓冲区连接到个报文段中这样可以通过最小化所发送报文数量来提高应用效率并减轻整体网络拥塞

由于这种算法对数据进行合并试图构成个完整TCP报文段因此会引入些延时Socket网络传输很长时间只发送些较小报文比如 telnet它让用户可以和远程系统进行交互通常通过个shell来进行如果用户被要求用发送报文的前输入来填充某个报文段该思路方法绝对不能满足需要再比如HTTP协议通常客户机浏览器会产生个小请求(条HTTP请求消息)然后Web服务器就会返回个更大响应(Web页面)最小化传输延时是首要在这种情况下Socket可以提供种解决方案即禁用Nagle算法可设置TCP\_NODELAY 选项TCP 禁用Nagle算法

使用Samba实验表明在服务器上Samba驱动器上读取数据时禁用Nagle算法几乎可以加倍提高读性能

#### 4.3 为Bandwidth Delay Product调节TCP窗口

TCP性能取决于几方面原因最重要是链接带宽(link bandwidth)(报文在网络上传输速率)和往返时间(round-trip time)或RTT(发送报文和接收到另一端响应的间延时)这两个值确定称为BDP(Bandwidth Delay Product)内容  
BDP给出种简单思路方法计算理论上最优TCP Socket缓冲区大小(其中保存排队等待传输和等待应用接收数据)缓冲区太小TCP窗口就不能完全打开这会限制性能;缓冲区太大则会浪费宝贵内存资源;设置缓冲区大小合适就可完全利用可用带宽

BDP计算公式:

$$BDP = \text{link bandwidth} \times RTT$$

若应用通过个100MB/s局域网通信其RTT为500ms则BDP为:50MB/s × 0.050/8625M = 625KBLinux 2.6默认TCP窗口大小是110KB这将连接带宽限制为22M/S计算思路方法如下:

$$\text{throughput} = \text{window\_size} / RTT$$

$$110 \text{ KB} / 0.050 = 2.2 \text{ MB/s}$$

使用上面计算窗口大小得到带宽为12.5 MB/s即:

$$625 \text{ KB} / 0.050 = 12.5 \text{ MB/s}$$

差别很大并且可以为Socket提供更大吞吐量可以根据自己Socket计算最优缓冲区大小Socket提供几个Socket选项其中两个可以用于修改Socket发送和接收缓冲区大小使用SO\_SNDBUF和SO\_RCVBUF选项来调整发送和接收缓冲区大小

在Linux 2.6内核中,发送缓冲区大小由用户定义而接收缓冲区会自动加倍通过计算合理设置缓冲区大小Socket网络传输带宽资源将得到充分利用从而提高了传输性能

#### 5 结束语

设计和实现个基于LinuxSocket网络编程通过在服务器端运行预先编译可执行文件serv和在客户端运行预先编译可执行文件clt服务器端和客户端建立通信连接加入select以后服务器端可以允许多个客户端接入服务器端解决了I / O多路复用问题更加接近实际应用利用TCP 禁用Nagle算法实现了最小化报文传输延时提高了Socket性能在网络带宽非常珍贵现实中提出了为Bandwidth Delay Product调节TCP窗口修改发送和接收缓冲区大小完全利用可用带宽达到较好网络传输效果实际网络传输环境复杂多变如何达到最理想网络传输还需进步分析和研究

2009-12-8 13:58:55

疯狂代码 <http://CrazyCoder.cn/>