

一台服务器可以发起多少个连接

li.w@ctrip.com

tcp	TIME-WAIT	0	0	10.8.110.73:39794	10.28.98.190:http
tcp	ESTAB	0	0	10.8.110.73:37801	10.28.98.215:http
tcp	TIME-WAIT	0	0	10.8.110.73:58559	10.8.44.8:http
tcp	ESTAB	0	0	10.8.110.73:37588	10.8.47.35:http
tcp	TIME-WAIT	0	0	10.8.110.73:55319	10.8.160.47:http
tcp	TIME-WAIT	0	0	10.8.110.73:28586	10.28.124.160:http
tcp	ESTAB	0	0	10.8.110.73:63381	10.8.106.66:http
tcp	TIME-WAIT	0	0	10.8.110.73:45392	10.8.47.48:http
tcp	TIME-WAIT	0	0	10.8.110.73:56374	10.8.47.101:http
tcp	TIME-WAIT	0	0	10.8.110.73:16940	10.28.82.90:http
tcp	TIME-WAIT	0	0	10.8.110.73:19321	10.28.96.238:http
tcp	ESTAB	0	0	10.8.110.73:49631	10.28.63.145:http
tcp	TIME-WAIT	0	0	10.8.110.73:17348	10.8.161.0:http
tcp	TIME-WAIT	0	0	10.8.110.73:64822	10.8.46.69:http
tcp	TIME-WAIT	0	0	10.8.110.73:43348	10.8.46.158:http
tcp	TIME-WAIT	0	0	10.8.110.73:13909	10.8.46.96:http
tcp	TIME-WAIT	0	0	10.8.110.73:62446	10.8.46.34:http
tcp	ESTAB	0	0	10.8.110.73:34966	10.28.63.133:http

65535 ?

动态端口限制 ?

Port already inuse

TCP连接四元组

- src_ip, src_port, dst_ip, dst_port
- 如果src_ip, src_port不变, 那变化的dst_ip, dst_port是不是可以建立更多的连接了?
 - tcp ESTAB 0 0 10.8.110.73:19985 10.8.46.200:http
 - tcp ESTAB 0 0 10.8.110.73:19985 10.8.47.13:http

REUSEADDR (UNPv1)

- ① 当有一个有相同本地地址和端口的socket1处于TIME_WAIT状态时，而你启动的程序的socket2要占用该地址和端口，你的程序就要用到该选项。
- ② SO_REUSEADDR允许同一port上启动同一服务器的多个实例(多个进程)。但每个实例绑定的IP地址是不能相同的。在有多块网卡或用IP Alias技术的机器可以测试这种情况。
- ③ SO_REUSEADDR允许单个进程绑定相同的端口到多个socket上，但每个socket绑定的ip地址不同。这和2很相似，区别请看UNPv1。
- ④ SO_REUSEADDR允许完全相同的地址和端口的重复绑定。但这只用于UDP的多播，不用于TCP。

DEMO

```
import socket

DST_HOST = '10.8.84.36'
DST_PORT = 80
SRC_PORT = 65249
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
s.bind(('', SRC_PORT))
s.connect((DST_HOST, DST_PORT))
```

- DST1:
 - 10.8.84.36 80
- DST2:
 - 10.9.196.63 80

```
[root@localhost ~]# netstat -antp | grep 65249
tcp        0      0 10.0.2.15:65249    10.9.196.63:80    ESTABLISHED 25220/python
tcp        0      0 10.0.2.15:65249    10.8.84.36:80     ESTABLISHED 25221/python
```

REUSEADDR (UNPv1)

- ① 当有一个有相同本地地址和端口的socket1处于TIME_WAIT状态时，而你启动的程序的socket2要占用该地址和端口，你的程序就要用到该选项。
- ② SO_REUSEADDR允许同一port上启动同一服务器的多个实例(多个进程)。但每个实例绑定的IP地址是不能相同的。在有多块网卡或用IP Alias技术的机器可以测试这种情况。
- ③ SO_REUSEADDR允许单个进程绑定相同的端口到多个socket上，但每个socket绑定的ip地址不同。这和2很相似，区别请看UNPv1。
- ④ SO_REUSEADDR允许完全相同的地址和端口的重复绑定。但这只用于UDP的多播，不用于TCP。

DEMO – 同一个进程

- DST1:
 - 140.207.228.45 80
- DST2:
 - 140.207.228.45 443

```
[root@localhost ~]# netstat -antp | grep 65249
tcp        0      0 10.0.2.15:65249    140.207.228.45:443 ESTABLISHED 25317/python
tcp        0      0 10.0.2.15:65249    140.207.228.45:80  ESTABLISHED 25317/python
```

REUSEADDR Client-side

SO_REUSEADDR

Indicates that the rules used in validating addresses supplied in a `bind(2)` call should allow reuse of local addresses. For **AF_INET** sockets this means that a socket may bind, except when there is an active listening socket bound to the address.

When the listening socket is bound to **INADDR_ANY** with a specific port then it is not possible to bind to this port for any local address. Argument is an integer boolean flag.

PRODUCT DEMO

- 一台tengine服务器可以往后端发起10w+连接

QA

- 一台服务器上的8080端口处于listen状态，同时打开了reuseaddr，这台服务器作为客户端访问其他服务器，在建连的时候也打开了reuseaddr，请问可以复用8080端口吗？

Ref & THANKS

- REUSEPORT&REUSEADDR:
 - <https://stackoverflow.com/questions/14388706/socket-options-so-reuseaddr-and-so-reuseport-how-do-they-differ-do-they-mean-t/14388707>
- Unix网络编程:
 - <https://github.com/yuyichao/cs-doc/blob/master/Programming/Linux/UNIX%E7%BD%91%E7%BB%9C%E7%BC%96%E7%A8%8B%E7%AC%AC%E4%B8%80%E5%8D%B7.pdf>
- <http://man7.org/linux/man-pages/man7/socket.7.html>