## TIME\_WAIT和CLOSE\_WAIT状态区别

<https://www.jianshu.com/p/f6a563f3526c?utm_source=oschina-app>

netstat -n | awk '/^tcp/ {++S[$NF]} END {for(a in S) print a, S[a]}'

常用的三个状态是：**ESTABLISHED** 表示正在通信，**TIME\_WAIT** 表示主动关闭，**CLOSE\_WAIT** 表示被动关闭。

* TCP协议规定，对于已经建立的连接，网络双方要进行四次握手才能成功断开连接，如果缺少了其中某个步骤，将会使连接处于假死状态，连接本身占用的资源不会被释放。网络服务器程序要同时管理大量连接，所以很有必要保证无用连接完全断开，否则大量僵死的连接会浪费许多服务器资源。在众多TCP状态中，最值得注意的状态有两个：**CLOSE\_WAIT和TIME\_WAIT。**

**TIME\_WAIT**

* TIME\_WAIT 是主动关闭链接时形成的，等待2MSL时间，约4分钟。主要是防止最后一个ACK丢失。 由于TIME\_WAIT 的时间会非常长，因此server端应尽量减少主动关闭连接

**CLOSE\_WAIT**

* CLOSE\_WAIT是被动关闭连接是形成的。根据TCP状态机，服务器端收到客户端发送的FIN，则按照TCP实现发送ACK，因此进入CLOSE\_WAIT状态。但如果服务器端不执行close()，就不能由CLOSE\_WAIT迁移到LAST\_ACK，则系统中会存在很多CLOSE\_WAIT状态的连接。此时，可能是系统忙于处理读、写操作，而未将已收到FIN的连接，进行close。此时，recv/read已收到FIN的连接socket，会返回0。

**为什么需要 TIME\_WAIT 状态？**

* 假设最终的ACK丢失，server将重发FIN，client必须维护TCP状态信息以便可以重发最终的ACK，否则会发送RST，结果server认为发生错误。TCP实现必须可靠地终止连接的两个方向(全双工关闭)，client必须进入 TIME\_WAIT 状态，因为client可能面 临重发最终ACK的情形。

**为什么 TIME\_WAIT 状态需要保持 2MSL 这么长的时间？**

* 如果 TIME\_WAIT 状态保持时间不足够长(比如小于2MSL)，第一个连接就正常终止了。第二个拥有相同相关五元组的连接出现，而第一个连接的重复报文到达，干扰了第二个连接。TCP实现必须防止某个连接的重复报文在连接终止后出现，所以让TIME\_WAIT状态保持时间足够长(2MSL)，连接相应方向上的TCP报文要么完全响应完毕，要么被 丢弃。建立第二个连接的时候，不会混淆。

### TIME\_WAIT 和CLOSE\_WAIT状态socket过多

**如果服务器出了异常，百分之八九十都是下面两种情况：**

* 1.服务器保持了大量TIME\_WAIT状态
* 2.服务器保持了大量CLOSE\_WAIT状态，简单来说CLOSE\_WAIT数目过大是由于被动关闭连接处理不当导致的。
* 因为linux分配给一个用户的文件句柄是有限的，而TIME\_WAIT和CLOSE\_WAIT两种状态如果一直被保持，那么意味着对应数目的通道就一直被占着，而且是“占着茅坑不使劲”，一旦达到句柄数上限，新的请求就无法被处理了，接着就是大量Too Many Open Files异常，Tomcat崩溃。