客户端非阻塞 socket 建链流程

TCP 协议是面向连接的、可靠的、基于字节流的传输层协议。那使用 tcp 协议进行通信的两端是如何进行通信的？使用 tcp 协议进行通信的两端是通过套接字（scoket）来建立连接的。套接字 socket 主要有两种类型，阻塞和非阻塞。通常为了防止进程阻塞以及避免 cpu 被长时间占用，客户端和服务端一般都会采用非阻塞 socket 进行通信，其中 Nginx 就是一个典型的例子。下面我们就以 Nginx 的 upstream 机制所涉及的与后端服务器建链的流程来总结下使用非阻塞 socket 的客户端建链流程。

先来看下 Nginx 的 upstream 机制所涉及的与后端服务器建链的流程，其相关函数为 ngx_event_connect_peer，该函数的执行流程如下：

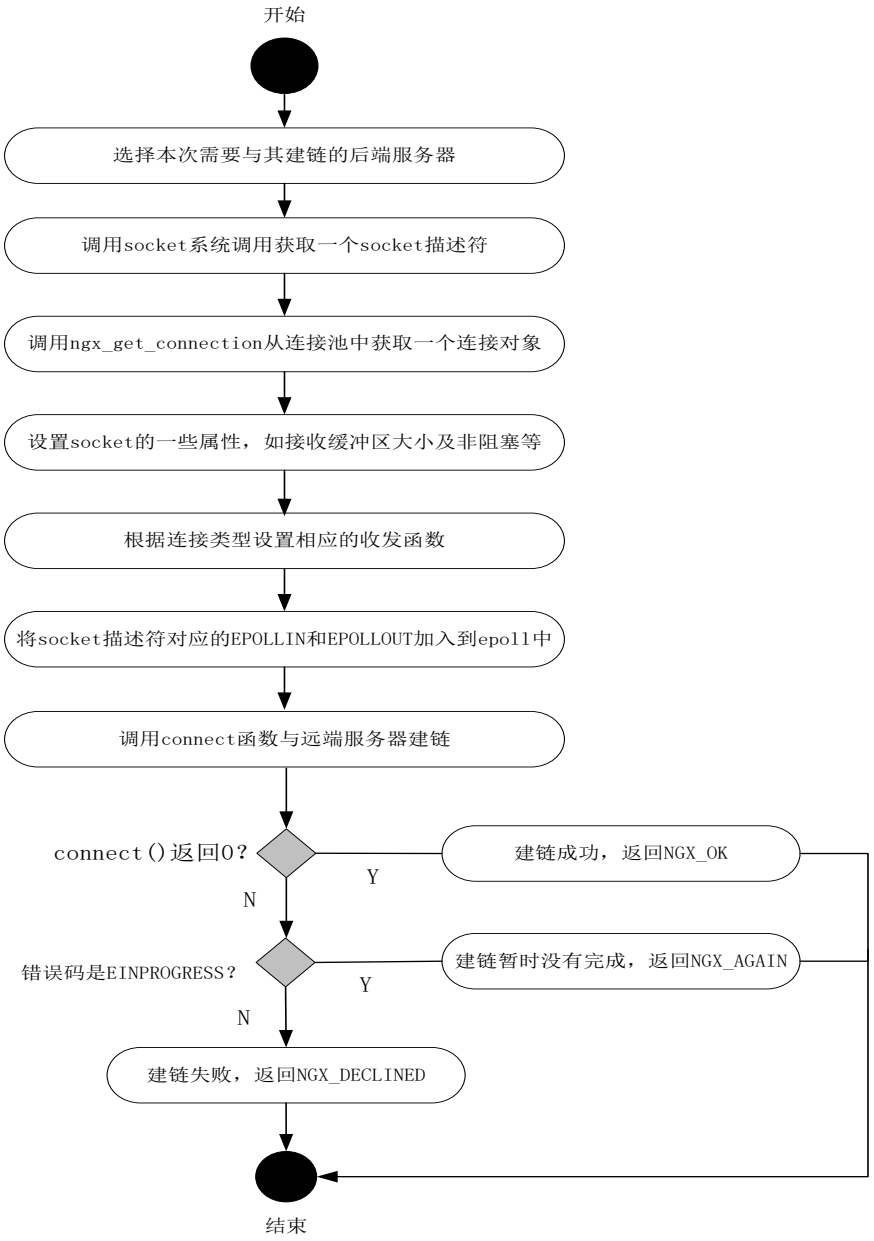

图 1 ngx_event_connect_peer。

在建链流程图中调用 connect 尝试与后端服务器建链那一步，如果 connect

返回 0 表明 Nginx 与后端服务器已经建立了连接，但是如果 connect 返回-1 并且错误码是 EINPROGRESS 表明后端服务器由于某些原因暂时没有完成连接的建立，后续建立连接后会通知 Nginx，此时 Nginx 的做法是，将连接对应的读写事件加入到了 epoll 中监控，并将写事件加入到定时器中，因为 Nginx 与后端服务器建立连接是有时间限制的，所以后续如果超时执行了写事件回调函数，则表示建链因超时而失败了。如果不是因为超时而是连接上有写事件发生调用事件回调函数，则会以 SO_ERROR 为参数调用 setsockopt 函数，然后根据错误码来判断 Nginx 是否与后端服务器建立了连接，这部分见在调用 ngx_event_connect_peer 函数的 ngx_stream_proxy_connect 函数中设置的读写事件回调函数 ngx_stream_proxy_connect_handler。

从上面 Nginx 的处理流程中我们可以看到非阻塞 socket 客户端建链的一般步骤：

1、 调用 socket 接口获取一个 socket 描述符。
2、 根据实际情况设置 socket 的一些属性，如接收缓冲区和发送缓冲区的大小等等。
3、 将 socket 设置为非阻塞 socket。通过调用哪个接口将 socket 设置为非阻塞的呢？可以以 FIONBIO 为参数调用 ioctl 函数将 socket 设置为非阻塞的，具体 ioctl 函数的描述和使用可以参见 Linux manual page。
4、 将 socket 描述符对应的 EPOLLIN 和 EPOLLOUT 事件加入到 epoll 的监控机制中，这样当 socket 描述符有事件发生时能够及时通知应用程序进行相应事件的处理。
5、 调用 connect 系统调用与后端服务器建立连接。如果 connect 函数返回 0 则表示应用程序与后端服务器建链成功，应用程序就可以执行下一步动作了；如果 connect 返回的是-1，那么就需要根据系统返回的错误码进一步分析导致调用 connect 出错的原因。在 Linux 下，如果错误码是 EINPROGRESS 表明与后端服务器只是暂时没有完成建链操作，并不代表失败，这个时候就需要设置 socket 描述符的 EPOLLIN 和 EPOLLOU 事件对应的处理函数。待后续 socket 写事件发生，epoll 就会通知应用程序该 socket 目前可写，然后调用写事件对应的处理函数，判断建链是否成功，如果建链成功，则应用程序就可以执行下一步操作了，如果建链失败，则返回。除了 EINPROGRESS 之外的错误码都表示 Nginx 和后端服务器建链失败了。

上面的第五步中对于 connect 的调用为什么需要做如此细化的处理呢？这个就要看非阻塞 socket 的特性了，因为对于非阻塞的 socket，connect 系统调用会立即返回，如果返回 0，表明建链应用程序与远端服务器建链成功，如果返回-1 且错误码是 EINPROGRESS，则会在后台进行三次握手的处理，后续可以通过 select、epoll 等 I/O 多路复用机制监控 socket 描述符的写事件来判断建链成功与否，在 connect()函数的 Linux manual page 中也有类似的描述，如下：

EINPROGRESS

The socket is nonblocking and the connection cannot be completed immediately. It is possible to select or poll for completion by selecting the socket for writing.After select indicates writability, use getsockopt to read the **SO_ERROR** option at level

SOL_SOCKET to determine whether **connect**() completed successfully (**SO_ERROR** is zero) or unsuccessfully (**SO_ERROR** is one of the usual error codes listed here, explaining the reason for the failure).