Nginx HTTP KeepAlive

# 概述

通过对TCP Connection的重用，可以避免频繁连接，提高性能。KeepAlive就是重用TCP Connection的机制。

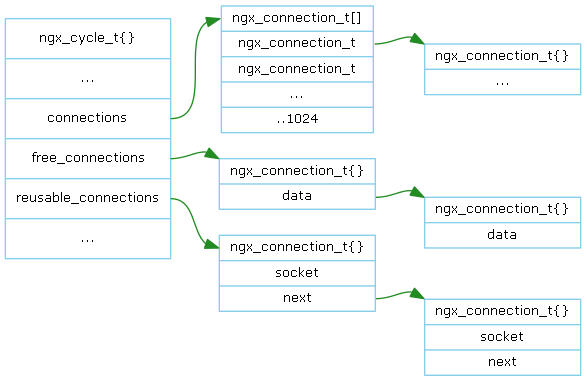
# Socket KeepAlive

TCP协议对TCP Socket KeepAlive提供了支持。在TCP Connection设一个Timer，当Timer触发时，发送一个数据长度为零需ACK的包，对端返回一个确认包。

在TCP Connection设另一个Timer，当这个Timer触发时，说明Connection已断，则close这个Connection。

<http://tldp.org/HOWTO/html_single/TCP-Keepalive-HOWTO/>

# Connections



ngx\_connection\_t是管理TCP Connection的数据结构。它在Nginx Worker进程启动时全部分配好资源，并由ngx\_cycle\_t管理。

* connections

所有的预分配的ngx\_connection\_t。

* free\_connections

空闲的ngx\_connection\_t（socket已关闭的）列表，可被直接使用。

* reusable\_connections

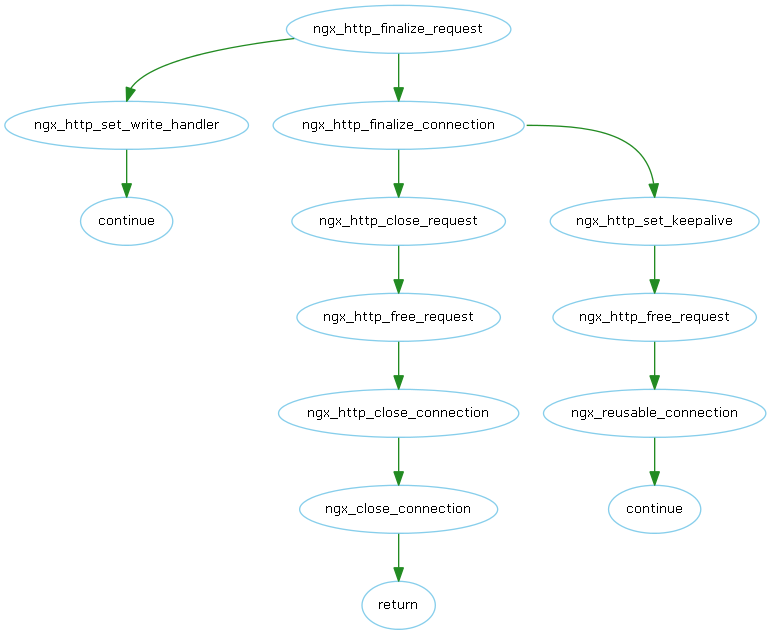
socket在工作event还在监听的ngx\_connection\_t，用于KeepAlive功能，当socket有数据到达时，回调函数即把它

从这个队列移除，开始正常处理请求。

当空闲的ngx\_connection\_t资源不足时，会关闭这个队列中若干个ngx\_connection\_t（不监听事件、关闭socket），

并加入free\_connections作为空闲资源使用。

# Nginx KeepAlive



## 无KeepAlive

如上图中线。

1. 释放ngx\_http\_request\_t、ngx\_buf\_t等内存资源。
2. 不监听Accepted Socket的读写事件。
3. 关闭ngx\_connection\_t中的Accepted Socket。
4. 加入空闲列表。

## KeepAlive

如上图右线。

1. 在ngx\_http\_finalize\_connection()过程中，设置读事件的回调函数为ngx\_http\_keepalive\_handler()。
2. 释放ngx\_http\_request\_t、ngx\_buf\_t等内存资源，以减少Memory Footprint。
3. 将ngx\_connection\_t加入可重用队列（不关闭Accepted Socket）。
4. 开始监听，监听到的读事件会是下列的三种情况，调用读事件回调函数ngx\_http\_keepalive\_handler()。
5. 新的请求到达
   1. 将ngx\_connection\_t从可重用队列移除。
   2. 调用ngx\_http\_init\_request()，此后过程同<Nginx HTTP Framework>。
6. 如果在规定时间内没有新的请求
   1. 不监听Accepted Socket的读写事件。
   2. 关闭ngx\_connection\_t中的Accepted Socket。
   3. 将ngx\_connection\_t从可重用队列移除。
   4. 加入空闲列表。
7. Client关闭连接。
   1. 不监听Accepted Socket的读写事件。
   2. 关闭ngx\_connection\_t中的Accepted Socket。
   3. 将ngx\_connection\_t从可重用队列移除。
   4. 加入空闲列表。

# Prototype

暂无

# Latest revision

https://github.com/lingjf/nginx\_analyse/blob/master/doc/