>    现在有一个问题是接口经常接口超时，因此想知道在哪个函数上延时比较严重，请在stap上有什么方法么？   
  
我们有一整套的基于 systemtap 的工具链可以用于在线或者离线的性能分析。   
  
你的 nginx 进程的 CPU 使用率如果很高的话，可以使用 C 级别的 on-CPU 时间火焰图工具对你最忙的 nginx worker 进程进行采样：   
  
    <https://github.com/agentzh/nginx-systemtap-toolkit#sample-bt>   
  
如果你的 nginx 进程的 CPU 很低，但请求延时很高，则有两种可能：   
  
1. 你的 nginx 阻塞在了某些阻塞的 IO 操作（比如文件 IO）或者系统的同步锁上，此时你可以使用 C 级别的 off-CPU   
时间火焰图工具对某个典型的 nginx worker 进程进行采样：   
  
   <https://github.com/agentzh/nginx-systemtap-toolkit#sample-bt-off-cpu>   
  
如果你发现 Lua 代码占用了大部分的 CPU 时间，则可以进一步使用 ngx-lua-exec-time 工具加以确认：   
  
    <https://github.com/agentzh/stapxx#ngx-lua-exec-time>   
  
进一步地，你可以使用 Lua 代码级别的 on-CPU 火焰图工具在 Lua 层面上分析 CPU 时间的分布。如果你使用的是 LuaJIT   
2.0.x，则可以使用下面这个工具进行采样：   
  
    <https://github.com/agentzh/nginx-systemtap-toolkit#ngx-sample-lua-bt>   
  
如果你使用的是 LuaJIT 2.1，则可以使用 lj-lua-stacks 工具进行采样：   
  
    <https://github.com/agentzh/stapxx#lj-lua-stacks>   
  
2. 你的 nginx 通过 ngx\_lua 的 cosocket 或者 ngx\_proxy 这样的 upstream   
模块和上游服务进行通信时，上游服务的延时过大。此时你可以分别使用   
ngx-lua-tcp-recv-time、ngx-lua-udp-recv-time 以及 ngx-single-req-latency   
工具进行分析：   
  
    <https://github.com/agentzh/stapxx#ngx-lua-tcp-recv-time>   
  
    <https://github.com/agentzh/stapxx#ngx-lua-udp-recv-time>   
  
    <https://github.com/agentzh/stapxx#ngx-single-req-latency>   
  
Best regards,   
-agentzh