黑客派

Nginx 的超时 timeout 配置详解

作者：shiguofu

原文链接：https://hacpai.com/article/1543809580828

来源网站：黑客派

<img src="https://static.hacpai.com/images/img-loading.svg" alt="WeChatScreenshot_20181203120213png" data-src="http://image.shiguofu.cn//file/2018/12/45c55789fea4454a8b6203607606f1e_WeChatScreenshot_20181203120213.png">
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>
<script>
    (adsbygoogle = window.adsbygoogle || []).push({});
</script>
本文介绍 Nginx 的 超时（timeout）配置。分享给大家，具体如下：
Nginx 处理的每个请求均有相应的超时设置。如果做好这些超时时间的限定，判定超时后资源被放，用来处理其他的请求，以此提升 Nginx 的性能。
keepalive_timeout
HTTP 是一种无状态协议，客户端向服务器发送一个 TCP 请求，服务端响应完毕后断开连接。
如果客户端向服务器发送多个请求，每个请求都要建立各自独立的连接以传输数据。
HTTP 有一个 KeepAlive 模式，它告诉 webserver 在处理完一个请求后保持这个 TCP 连接的打状态。若接收到来自客户端的其它请求，服务端会利用这个未被关闭的连接，而不需要再建立一个连。
KeepAlive 在一段时间内保持打开状态，它们会在这段时间内占用资源。占用过多就会影响性能
Nginx 使用 keepalive_timeout 来指定 KeepAlive 的超时时间（timeout）。指定每个 TCP 连最多可以保持多长时间。Nginx 的默认值是 75 秒，有些浏览器最多只保持 60 秒，所以可以设定为 6 秒。若将它设置为 0，就禁止了 keepalive 连接。通常 keepalive_timeout 应该比 client_body_timout(见下文)大。
<blockquote>
 <code># 配置段: http, server, location</code>
</blockquote>
<blockquote>
 <code>keepalive_timeout 60s;</code>
</blockquote>
client_body_timeout
指定客户端与服务端建立连接后发送 request body 的超时时间。如果客户端在指定时间内没有送任何内容，Nginx 返回 HTTP 408（Request Timed Out）。
<blockquote>
 <code># 配置段: http, server, location</code>
</blockquote>
<blockquote>
 <code>client_body_timeout 20s;</code>
</blockquote>
client_header_timeout
客户端向服务端发送一个完整的 request header 的超时时间。如果客户端在指定时间内没有发一个完整的 request header，Nginx 返回 HTTP 408（Request Timed Out）。
<blockquote>
 <code># 配置段: http, server, location</code>
</blockquote>
<blockquote>
 <code>client_header_timeout 10s;</code>
</blockquote>
send_timeout
服务端向客户端传输数据的超时时间，根据转发的应用服务可以配置 proxy_send_timeout、uw

gi_send_timeout、fastcgi_send_timeout（见下文）。
<blockquote>
<code># 配置段:http, server, location</code>
</blockquote>
<blockquote>
<code>send_timeout 30s;</code>
 <code>Default: </code>
<code>send_timeout 60s;</code>
<code>Context: http, server, location</code>
<code>Sets a timeout for transmitting a response to the client. The timeout is set only between two successive write operations,</code>
<code>not for the transmission of the whole response. If the client does not receive anything within this time, the connection is closed.</code>
</blockquote>

client_header_timeout
接收客户端 header 超时， 默认 60s, 如果 60s 内没有收到完整的 http 包头， 返回 408
<blockquote>
<code>Syntax: client_header_timeout time;</code>
</blockquote>
<blockquote>
<code>Default: </code>
</blockquote>
<blockquote>
<code>client_header_timeout 60s;</code>
</blockquote>
<blockquote>
<code>Context: http, server</code>
</blockquote>
<blockquote>
<code>Defines a timeout for reading client request header. If a client does not transmit the entire header within this time,</code>
</blockquote>
<blockquote>
<code>the 408 (Request Time-out) error is returned to the client.</code>
</blockquote>
client_body_timeout
接收客户端 body 超时， 默认 60s, 如果连续的 60s 内没有收到客户端的 1 个字节， 返回 408
<blockquote>
<code>Syntax: client_body_timeout time;</code>
</blockquote>
<blockquote>
<code>Default: </code>
</blockquote>
<blockquote>
<code>client_body_timeout 60s;</code>

</blockquote>
<blockquote>
<code>Context: http, server, location</code>
</blockquote>
<blockquote>
<code>Defines a timeout for reading client request body. The timeout is set only for a p
riod between two successive read operations, not for the transmission of the whole request b
dy.</code>
</blockquote>
<blockquote>
<code>If a client does not transmit anything within this time,</code>
</blockquote>
<blockquote>
<code>the 408 (Request Time-out) error is returned to the client.</code>
</blockquote>
lingering_timeout
可以理解为 TCP 连接关闭时的 SO_LINGER 延时设置，默认 5s
<blockquote>
<code>Syntax: lingering_timeout time;</code>
</blockquote>
<blockquote>
<code>Default: </code>
</blockquote>
<blockquote>
<code>lingering_timeout 5s;</code>
</blockquote>
<blockquote>
<code>Context: http, server, location</code>
</blockquote>
<blockquote>
<code>When lingering_close is in effect, this directive specifies the maximum waiting ti
e for more client data to arrive. If data are not received during this time,</code>
</blockquote>
<blockquote>
<code>the connection is closed. Otherwise, the data are read and ignored, and nginx st
rts waiting for more data again.</code>
</blockquote>
<blockquote>
<code>The “wait-read-ignore” cycle is repeated, but no longer than specified by the l
ngering_time directive.</code>
</blockquote>

resolver_timeout
域名解析超时，默认 30s
<blockquote>
<code>Syntax: resolver_timeout time;</code>

<code>Default: </code>
<code>resolver_timeout 30s;</code>
<code>Context: http, server, location</code>
<code>Sets a timeout for name resolution, for example:</code>
<code>resolver_timeout 5s;</code>
</blockquote>
proxy_connect_timeout
nginx 与 upstream server 的连接超时时间，默认为 60s；根据应用不同可配置 uwsgi_send_timeout/fascgi_send_timeout/proxy_send_timeout
<blockquote>
<code>Syntax: proxy_connect_timeout time;</code>
<code>Default: </code>
<code>proxy_connect_timeout 60s;</code>
<code>Context: http, server, location</code>
<code>Defines a timeout for establishing a connection with a proxied server. It should e noted that this timeout cannot usually exceed 75 seconds.</code>
</blockquote>
proxy_read_timeout
nginx 接收 upstream server 数据超时， 默认 60s, 如果连续的 60s 内没有收到 1 个字节， 连关闭；根据应用不同可配置 uwsgi_send_timeout/fascgi_send_timeout/proxy_send_timeout

<blockquote>
<code>Syntax: proxy_read_timeout time;</code>
<code>Default: </code>
<code>proxy_read_timeout 60s;</code>
<code>Context: http, server, location</code>
<code>Defines a timeout for reading a response from the proxied server. The timeout is set only between two successive read operations,</code>
<code>not for the transmission of the whole response. If the proxied server does not transmit anything within this time, the connection is closed.</code>
</blockquote>
proxy_send_timeout
nginx 发送数据至 upstream server 超时， 默认 60s, 如果连续的 60s 内没有发送 1 个字节，接关闭；根据应用不同可配置 uwsgi_send_timeout/fascgi_send_timeout/proxy_send_timeout。/p>
<blockquote>
<code>Syntax: proxy_send_timeout time;</code>
<code>Default: </code>
<code>proxy_send_timeout 60s;</code>
<code>Context: http, server, location</code>
<code>Sets a timeout for transmitting a request to the proxied server. The timeout is set only between two successive write operations,</code>
<code>not for the transmission of the whole request. If the proxied server does not rec/code>
</blockquote>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in>
<script>
    (adsbygoogle = window.adsbygoogle || []).push({});
</script>