**Nginx核心要领五：worker\_processes、worker\_connections设置**

worker\_processes与worker\_connections 设置好合适大小，可以提示nginx处理性能，非常重要。

原作者的话：

As a general rule you need the only worker with large number of worker\_connections, say 10,000 or 20,000.

However, if nginx does CPU-intensive work as SSL or gzipping and you have 2 or more CPU, then you may

set worker\_processes to be equal to CPU number.

Besides, if you serve many static files and the total size of the files is bigger than memory, then you may

increase worker\_processes to utilize a full disk bandwidth.

Igor Sysoev

翻译：

一般一个进程足够了，你可以把连接数设得很大。（worker\_processes: 1，worker\_connections: 10,000）

如果有SSL、gzip这些比较消耗CPU的工作，而且是多核CPU的话，可以设为和CPU的数量一样。(worker\_processes: CPU核心数)

或者要处理很多很多的小文件，而且文件总大小比内存大很多的时候，也可以把进程数增加，以充分利用IO带宽（主要似乎是IO操作有block）

**worker\_processes，工作进程数**

* 1.默认：worker\_processes: 1
* 2.调大：worker\_processes: CPU核心数，(双核4线程，可以设置为4)

**worker\_connections，单个工作进程可以允许同时建立外部连接的数量**

数字越大，能同时处理的连接越多

* 1.默认：worker\_connections: 1024
* 2.调大：worker\_connections: 100000，（调大到10万连接）

**worker\_connections解析**

* 1.connections不是随便设置的，而是与两个指标有重要关联，一是内存，二是操作系统级别的“进程最大可打开文件数”。
* 2.内存：每个连接数分别对应一个read\_event、一个write\_event事件，一个连接数大概占用232字节，2个事件总占用96字节，那么一个连接总共占用328字节，通过数学公式可以算出100000个连接数大概会占用 31M = 100000 \* 328 / 1024 / 1024，当然这只是nginx启动时，connections连接数所占用的nginx。
* 3.进程最大可打开文件数：进程最大可打开文件数受限于操作系统，可通过 ulimit -n 命令查询，以前是1024，现在是65535,  
  nginx提供了worker\_rlimit\_nofile指令，这是除了ulimit的一种设置可用的描述符的方式。 该指令与使用ulimit对用户的设置是同样的效果。此指令的值将覆盖ulimit的值，如：worker\_rlimit\_nofile 20960;  
  设置ulimits：ulimit -SHn 65535

worker\_processes 2;

worker\_rlimit\_nofile 65535;

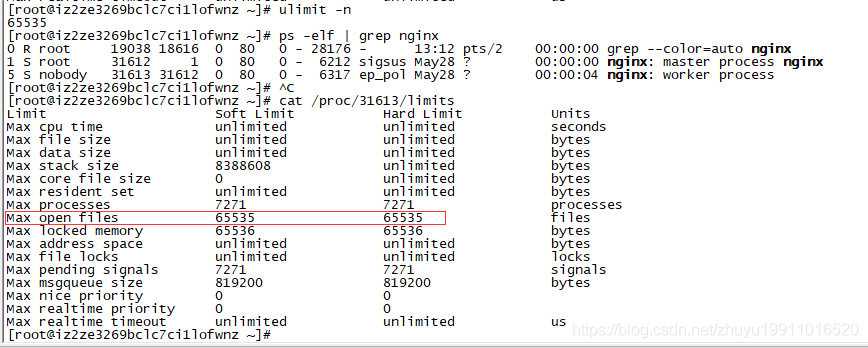
#pid logs/nginx.pid;

events {

worker\_connections 65535;

}

通过 ps -elf | grep nginx 找到 nginx 的worker进程ID  
通过 cat /proc/31613/limits 查看，其中2291是worker进程ID，请注意其中的Max open files

  
从上面的分析可以看出nginx占用内存小，处理性能高，通过提高服务器的配置，Nginx可以应对更大的连接数