NGINX 缓存使用指南

Nginx

一个web缓存坐落于客户端和“原始服务器（origin server）”中间，它保留了所有可见内容的拷贝。如果一个客户端请求的内容在缓存中存储，则可以直接在缓存中获得该内容而不需要与服务器通信。这样一来，由于web缓存距离客户端“更近”，就可以提高响应性能，并更有效率的使用应用服务器，因为服务器不用每次请求都进行页面生成工作。
在浏览器和应用服务器之间，存在多种“潜在”缓存，如：客户端浏览器缓存、中间缓存、内容分发网络（CDN）和服务器上的负载平衡和反向代理。缓存，仅在反向代理和负载均衡的层面，就对性能提高有很大的帮助。

Nginx从0.7.48版本开始，支持了类似Squid的缓存功能。这个缓存是把URL及相关组合当作Key，用md5编码哈希后保存在硬盘上，所以它可以支持任意URL链接，同时也支持404/301/302这样的非200状态码。虽然目前官方的Nginx Web缓存服务只能为指定URL或状态码设置过期时间，不支持类似Squid的PURGE指令，手动清除指定缓存页面，但是，通过一个第三方的Nginx模块，可以清除指定URL的缓存。

Nginx的Web缓存服务主要由proxy_cache相关指令集和fastcgi_cache相关指令集构成，前者用于反向代理时，对后端内容源服务器进行缓存，后者主要用于对FastCGI的动态程序进行缓存。两者的功能基本上一样。

最新的Nginx 0.8.32版本，proxy_cache和fastcgi_cache已经比较完善，加上第三方的ngx_cache_purge模块（用于清除指定URL的缓存），已经可以完全取代Squid。我们已经在生产环境使用了 Nginx 的 proxy_cache 缓存功能超过两个月，十分稳定，速度不逊于 Squid。

在功能上，Nginx已经具备Squid所拥有的Web缓存加速功能、清除指定URL缓存的功能。而在性能上，Nginx对多核CPU的利用，胜过Squid不少。另外，在反向代理、负载均衡、健康检查、后端服务器故障转移、Rewrite重写、易用性上，Nginx也比Squid强大得多。这使得一台Nginx可以同时作为“负载均衡服务器”与“Web缓存服务器”来使用。

一、如何安装和配置基础缓存

我们只需要两个命令就可以启用基础缓存： proxy_cache_path和 proxy_cache
proxy_cache_path用来设置缓存的路径和配置，proxy_cache用来启用缓存。

```
1. proxy_cache_path /path/to/cache levels=1:2
   keys_zone=my_cache:10m max_size=10g inactive=60m
2. use_temp_path=off;
3. server {
4. ...
5. location / {
6. proxy_cache my_cache;
7. proxy_pass http://my_upstream;
8. }
9. }
```

proxy_cache_path命令中的参数及对应配置说明如下：

> 1.用于缓存的本地磁盘目录是/path/to/cache/
>
> 2.levels在/path/to/cache/设置了一个两级层次结构的目录。将大量的文件放置在单个目录中会导致文件访问缓慢，所以针对大多数部署，我们推荐使用两级目录层次结构。如果levels参数没有配置，则NGINX会将所有的文件放到同一个目录中。
>
> 3.keys_zone设置一个共享内存区，该内存区用于存储缓存键和元数据，有些类似计时器的用途。将键的拷贝放入内存可以使NGINX在不检索磁盘的情况下快速决定一个请求是HIT还是MISS，这样大大提高了检索速度。一个1MB的内存空间可以存储大约8000个key，那么上面配置的10MB内存空间可以存储差不多80000个key。
>
> 4.max_size设置了缓存的上限（在上面的例子中是10G）。这是一个可选项；如果不指定具体值，那就是允许缓存不断增长，占用所有可用

的磁盘空间。当缓存达到这个上线，处理器便调用cache manager来移除最近最少被使用的文件，这样把缓存的空间降低至这个限制之下。

5.inactive指定了项目在不被访问的情况下能够在内存中保持的时间。在上面的例子中，如果一个文件在60分钟之内没有被请求，则缓存管理将会自动将其在内存中删除，不管该文件是否过期。该参数默认值为10分钟（10m）。注意，非活动内容有别于过期内容。NGINX不会自动删除由缓存控制头部指定的过期内容（本例中Cache-Control:max-age=120）。过期内容只有在inactive指定时间内没有被访问的情况下才会被删除。如果过期内容被访问了，那么NGINX就会将其从原服务器上刷新，并更新对应的inactive计时器。

6.NGINX最初会将注定写入缓存的文件先放入一个临时存储区域，use_temp_path=off命令指示NGINX将在缓存这些文件时将它们写入同一个目录下。我们强烈建议你将参数设置为off来避免在文件系统中不必要的数据拷贝。use_temp_path在NGINX1.7版本和NGINX Plus R6中有所介绍。

最终， proxy_cache命令启动缓存那些URL与location部分匹配的内容（本例中，为/）。你同样可以将proxy_cache命令添加到server部分，这将会将缓存应用到所有的那些location中未指定自己的proxy_cache命令的服务中。

二、陈旧总比没有强

NGINX内容缓存的一个非常强大的特性是：当无法从原始服务器获取最新的内容时，NGINX可以分发缓存中的陈旧（stale，编者注：即过期内容）内容。这种情况一般发生在关联缓存内容的原始服务器宕机或者繁忙时。比起对客户端传达错误信息，NGINX可发送在其内存中的陈旧的文件。NGINX的这种代理方式，为服务器提供额外级别的容错能力，并确保了在服务器故障或流量峰值的情况下的正常运行。为了开启该功能，只需要添加proxy_cache_use_stale命令即可：

```
1. location / {
2. ...
```

```
3. proxy_cache_use_stale error timeout http_500 http_502
   http_503 http_504;
4. }
```

按照上面例子中的配置，当NGINX收到服务器返回的error，timeout或者其他指定的5xx错误，并且在其缓存中有请求文件的陈旧版本，则会将这些陈旧版本的文件而不是错误信息发送给客户端。

三、缓存微调

NGINX提供了丰富的可选项配置用于缓存性能的微调。下面是使用了几个配置的例子：

```
1. proxy_cache_path /path/to/cache levels=1:2
   keys_zone=my_cache:10m max_size=10g inactive=60m
2. use_temp_path=off;
3.
4. server {
5. ...
6. location / {
7. proxy_cache my_cache;
8. proxy_cache_revalidate on;
9. proxy_cache_min_uses 3;
10. proxy_cache_use_stale error timeout updating http_500
    http_502 http_503 http_504;
11. proxy_cache_lock on;
12.
13. proxy_pass http://my_upstream;
14. }
15. }
```

配置解释如下：

> 1. proxy_cache_revalidate指示NGINX在刷新来自服务器的内容时使用GET请求。如果客户端的请求项已经被缓存过了，但是在缓存

控制头部中定义为过期，那么NGINX就会在GET请求中包含If-Modified-Since字段，发送至服务器端。这项配置可以节约带宽，因为对于NGINX已经缓存过的文件，服务器只会在该文件请求头中Last-Modified记录的时间内被修改时才将全部文件一起发送。

2. proxy_cache_min_uses设置了在NGINX缓存前，客户端请求一个条目的最短时间。当缓存不断被填满时，这项设置便十分有用，因为这确保了只有那些被经常访问的内容才会被添加到缓存中。该项默认值为1。

3. proxy_cache_use_stale中的updating参数告知NGINX在客户端请求的项目的更新正在原服务器中下载时发送旧内容，而不是向服务器转发重复的请求。第一个请求陈旧文件的用户不得不等待文件在原服务器中更新完毕。陈旧的文件会返回给随后的请求直到更新后的文件被全部下载。

4. 当proxy_cache_lock被启用时，当多个客户端请求一个缓存中不存在的文件（或称之为一个MISS），只有这些请求中的第一个被允许发送至服务器。其他请求在第一个请求得到满意结果之后在缓存中得到文件。如果不启用proxy_cache_lock，则所有在缓存中找不到文件的请求都会直接与服务器通信。

四、NGINX 如何决定是否缓存?

默认情况下，NGINX需要考虑从原始服务器得到的Cache-Control标头。当在响应头部中Cache-Control被配置为Private，No-Cache，No-Store或者Set-Cookie，NGINX不进行缓存。NGINX仅仅缓存GET和HEAD客户端请求。你也可以参照下面的解答覆盖这些默认值。

1、Cache-Control头部可否被忽略?
可以，使用proxy_ignore_headers命令。如下列配置：

```
1. ocation /images/ {
2. proxy_cache my_cache;
3. proxy_ignore_headers Cache-Control;
4. proxy_cache_valid any 30m;
```

5. ...
6. }

NGINX会忽略所有/images/下的Cache-Control头。proxy_cache_valid命令强制规定缓存数据的过期时间，如果忽略Cache-Control头，则该命令是十分必要的。NGINX不会缓存没有过期时间的文件。

2、NGINX能否缓存POST 请求?
可以，使用proxy_cache_methods命令：

```
1. proxy_cache_methods GET HEAD POST;
```

3、NGINX 可以缓存动态内容吗?

> 可以，提供的Cache-Control头部可以做到。缓存动态内容，甚至短时间内的内容可以减少在原始数据库和服务器中加载，可以提高第一个字节的到达时间，因为页面不需要对每个请求都生成一次。

4、可以使用Cookie作为缓存键的一部分吗?
可以，缓存键可以配置为任意值，如：

```
1. proxy_cache_key $proxy_host$request_uri$cookie_jessionid;
```

5、NGINX 如何处理字节范围请求?

> 如果缓存中的文件是最新的，NGINX会对客户端提出的字节范围请求传递指定的字节。如果文件并没有被提前缓存，或者是陈旧的，那么NGINX会从服务器上下载完整文件。如果请求了单字节范围，NGINX会尽快的将该字节发送给客户端，如果在下载的数据流中刚好有这个字节。如果请求指定了同一个文件中的多个字节范围，NGINX则会在文件下载完毕时将整个文件发送给客户端。 一旦文件下载完毕，NGINX将整个数据移动到缓存中，这样一来，无论将来的字节范围请求是单字节还是多字节范围，NGINX都可以在缓存中找到指定的内容立即响应。

参考：
NGINX其他优化

NGINX缓存使用官方指南

使用Nginx的proxy_cache缓存功能取代Squid[原创]

Nginx proxy cache官方解释

为Nginx安装清缓存模块——优化nginx服务器性能