讲师介绍

Hash　　QQ：805921455

从事Java软件研发近十年。

前新浪支付核心成员、

咪咕视讯(中国移动)项目经理、

对分布式架构、高性能编程有深入的研究。

明天，你一定会感谢今天奋力拼搏的自己

电商高并发缓存实战之通过Nginx缓存你的电商数据

分布式高并发—负载均衡

目录

课程安排

01

Nginx缓存机制介绍

Nginx中如何使用缓存？缓存示例

02

缓存清除机制分析

Nginx中缓存如何管理清除？

03

第三方缓存主动清除

第三方模块来进行主动缓存清楚

04

总结

本堂知识点整理

目录

Nginx缓存机制介绍

Nginx缓存机制的作用

1. 缓存能够提升性能，学会Nginx中如何使用缓存很重要。

2. Nginx作为静态资源服务器，静态资源变动频率小，缓存能够加速访问。

3. Nginx离用户最近，启用缓存能更好的提高性能，结合Redis可以组成类似二级缓存。

nginx缓存机制简述

Nginx中的缓存

nginx中的缓存是以文件系统上的分层数据存储的形式实现的。

缓存Key是可配置的，并且可以使用不同的请求特定参数来控制进入缓存的内容。

缓存Key和缓存元数据存储在共享内存段中，缓存加载器、缓存管理器和Worker进程可以访问它们。

目前，除了操作系统的虚拟文件系统机制所暗示的优化之外，没有任何内存中的文件缓存。每个缓存的响应都放在文件系统上的不同文件中。通过nginx配置指令控制指定层次结构(级别和命名细节)。

缓存放置内容过程：

当nginx从上游服务器读取响应时，首先将内容写入缓存目录结构之外的临时文件。

当nginx完成处理请求时，它会重命名临时文件并将其移动到缓存目录。如果用于代理的临时文件目录位于另一个文件系统上，则文件将被复制，因此建议将临时目录和缓存目录保留在同一个文件系统上。当需要显式清除文件时，从缓存目录结构中删除文件也是非常安全的。

nginx有第三方扩展，可以远程控制缓存的内容，并计划在主发行版中集成此功能

Nginx缓存支持

Nginx中有非常强大的缓存功能，针对后台服务返回的数据能够进行缓存，再次访问时，无需从后台服务拿取结果，直接在nginx本地获取即可。Nginx中针对fastcgi、http_proxy、scgi、ssl_session、ngx_http_uwsgi_module模块提供了通用的缓存功能。

cache支持的模块	备注
ngx_http_fastcgi_module	快速通用网关接口，如php,perl,tcl等采用
ngx_http_proxy_module	http代理模块，适用任何http协议的后台服务
ngx_http_scgi_module	通用网关接口，如php,perl,tcl等采用
ngx_http_uwsgi_module	WSGI协议的web服务，如python采用
ngx_http_ssl_module	session缓存
ngx_http_core_module	open_file_cache文件缓存

以http_proxy模块为例来进行讲解，观察官网其他模块，都相差不大，学会一个其他的也就学会了

缓存使用

缓存常用指令：

指令	作用	默认值	分类
proxy_cache_path	定义缓存的路径和缓存空间名、大小等配置，缓存数据存储在文件中。缓存中的文件名是将MD5功能应用于缓存键的结果。	-	定义
proxy_cache	启用缓存，指定的用于缓存的缓存空间名，不同地方可以启用同一个空间名。	-	应用
proxy_cache_valid	设置不同响应代码的缓存时间	-	应用
proxy_cache_key	定义缓存的Key	$scheme$proxy_host$request_uri;	定义
proxy_cache_purge	定义将请求视为缓存清除请求的条件。如果字符串参数的至少一个值不为空且不等于“0”，则移除具有相应高速缓存键的高速缓存条目 。通过返回204（无内容）响应来指示成功操作的结果。	-	应用

示例参考《nginx_cache.conf》文件

proxy_cache_path参数详解

level

用来定义缓存的层级，可以定义1到3个层级，每个层级接收值1、2

use_temp_path

是否启用缓存临时文件，第一次响应的内容将写入临时文件，后面才会重写回来。使用网盘注意网络IO的开销。

keys_zone

定义在共享存储区中存储了所有活跃的Key和关于数据的信息。存储区名称和大小由key_zone参数配置。一个兆字节的区域可以存储大约8,000个Key

Inactive

不论数据新旧程度，在inactive指定的时间内未访问的缓存数据 将从缓存中删除。

缓存指令附录一

指令	作用	默认值	分类
proxy_cache	定义用于缓存的共享内存区域，可以在多个地方使用相同的区域。off禁用从先前配置级别继承的高速缓存。	off	定义
proxy_cache_background_update	允许启动后台子请求以更新过期的缓存项，同时将过时的缓存响应返回给客户端。	off	更新
proxy_cache_bypass	定义不从缓存中获取响应的条件。如果字符串参数的至少一个值不为空且不等于“0”，则不会从缓存中获取响应，可以与proxy_no_cache指令一起使用	-	应用
proxy_cache_convert_head	启用或禁用将“ HEAD”方法转换为“ ”以GET进行缓存。禁用转换时， 应将缓存键配置为包含$request_method。	on	应用
proxy_cache_key	定义缓存的键，默认情况下，指令的值接近字符串	$ scheme $ proxy_host $ request_uri	定义
proxy_cache_lock	启用锁，一次只允许一个请求重建缓存，其他请求等待直到proxy_cache_lock_timeout设置超时。	off	管理
proxy_cache_lock_age	在指定的时间内，上一个重构缓存的请求未完成，则将另一个请求传递给代理服务器。	5s	应用
proxy_cache_lock_timeout	设置proxy_cache_lock的超时，超时请求将被传递给代理的服务器，但是，响应不会被缓存。	5s	应用
proxy_cache_max_range_offset	设置响应被缓存的最大字节数，超过则不缓存响应，直接请求代理服务。	-	应用
proxy_cache_methods	需要进行缓存的HTTP方法	GET、HEAD	应用
proxy_cache_min_uses	设置多少次请求后才缓存响应内容	1	应用

缓存指令附录二

指令	作用	默认值	分类
proxy_cache_path	设置缓存的路径和其他参数，缓存数据存储在文件中。缓存中的文件名是将MD5功能应用于缓存键的结果。 其他参数 levels参数定义高速缓存的层次结构级别 use_temp_path是否启用临时文件 keys_zone定义键的大小，一兆字节区域可以存储大约8000个键 inactive 指定的时间内未访问的缓存数据 将从缓存中删除 缓存管理器：max_size、manager_files、manager_threshold、manager_sleep 缓存清除：purger、purger_files、purger_threshold、purger_sleep	-	定义/清除
proxy_cache_purge	定义将请求视为缓存清除请求的条件。如果字符串参数的至少一个值不为空且不等于“0”，则移除具有相应高速缓存键的高速缓存条目 。通过返回204（无内容）响应来指示成功操作的结果。 如果清除请求的缓存键以星号（“ *”）结束，则将从缓存中删除与通配符键匹配的所有缓存条目。但是，这些条目将保留在磁盘上，直到它们被删除为非活动状态，或由缓存清除程序（1.7.12）处理，或者客户端尝试访问它们。	-	清除
proxy_cache_revalidate	使用带有“If-Modified-Since”和“If-None-Match”标头字段的条件请求启用过期缓存项的重新验证。	off	应用
proxy_cache_use_stale	确定在与代理服务器通信期间，可以在哪些情况下使用过时的缓存响应。该指令的参数与proxy_next_upstream指令的参数匹配	off	应用
proxy_cache_valid	设置不同响应代码的缓存时间	-	应用
proxy_no_cache	定义不将响应保存到缓存的条件。如果字符串参数的至少一个值不为空且不等于“0”，则不会保存响应。	-	应用

目录

缓存清除机制分析

被动缓存清除

proxy_cache_path指令的缓存管理

proxy_cache_path中可以通过以下指令来管理缓存

max_size

指定缓存大小，缓存管理进程监控缓存是否超过指定值，超过该大小则通过LRU算法来淘汰数据。一次迭代删除的数据通过下面的参数来指定。

manager_files

一次迭代过程中删除的项的数量，默认100个

manager_threshold

一次迭代操作的持续时间限制，默认200毫秒

manager_sleep

两次迭代的间隔时间，默认50毫秒

缓存加载

proxy_cache_path中可以通过以下参数来调整加载缓存

Nginx启动一分钟后，缓存加载进程被激活，存储在文件系统上先前缓存的数据将被加载到缓存区中，整个加载是在迭代中完成的。一次加载

loader_files

一次迭代加载不超过指定数目的项，默认100。

loader_threshold

一次迭代操作的持续时间限制，默认200毫秒

loader_sleep

两次迭代的间隔时间，默认50毫秒

主动清除缓存

Nginx商业功能

proxy_cache_path中可以通过以下参数来调整主动清除缓存

purger：on开启缓存清除进程，遍历所有缓存条目并删除匹配到的键的缓存数据

purger_files

一次迭代过程中扫描的项的数量，默认10个

purger_threshold

一次迭代的持续时间，默认50毫秒

purger_sleep

两次迭代的间隔时间，默认50毫秒

以上参数与proxy_cache_purge指令配合进行。

虽说只能在商业版的Nginx中使用，但是我们可以通过强大的第三方模块来替代

目录

第三方缓存主动清除

ngx_cache_purge

ngx_cache_purge是一个第三方的nginx缓存主动清除模块，集成方便，使用简单。

```
wget https://github.com/FRiCKLE/ngx_cache_purge/archive/2.3.tar.gz
tar -xfvz ngx_cache_purge-2.3.tar.gz
./configure --prefix=/app/nginx --with-http_stub_status_module --with-
http_ssl_module --add-module=../ngx_cache_purge-2.3
make
make install
```

详细资料地址：https://github.com/FRiCKLE/ngx_cache_purge

使用ngx_cache_purge

1. 访问缓存URL，http://hostname/test/n.jpg

   服务端初次响应，建立缓存内容

2. 查看缓存的文件

   查看缓存文件系统内容

3. 修改数据访问

   修改数据以后，缓存还在起作用，一段时间不能访问到新数据

4. 清除缓存再次访问

   通过插件主动清除缓存，再次访问到新的数据内容

目录

总结

总结

Nginx缓存支持的模块和常用指令

Nginx中缓存管理机制

- 缓存开启
- 缓存管理
- 缓存加载
- 缓存清除

第三方缓存主动清除插件

- 安装使用
- 测试验证

谢谢观看