高并发

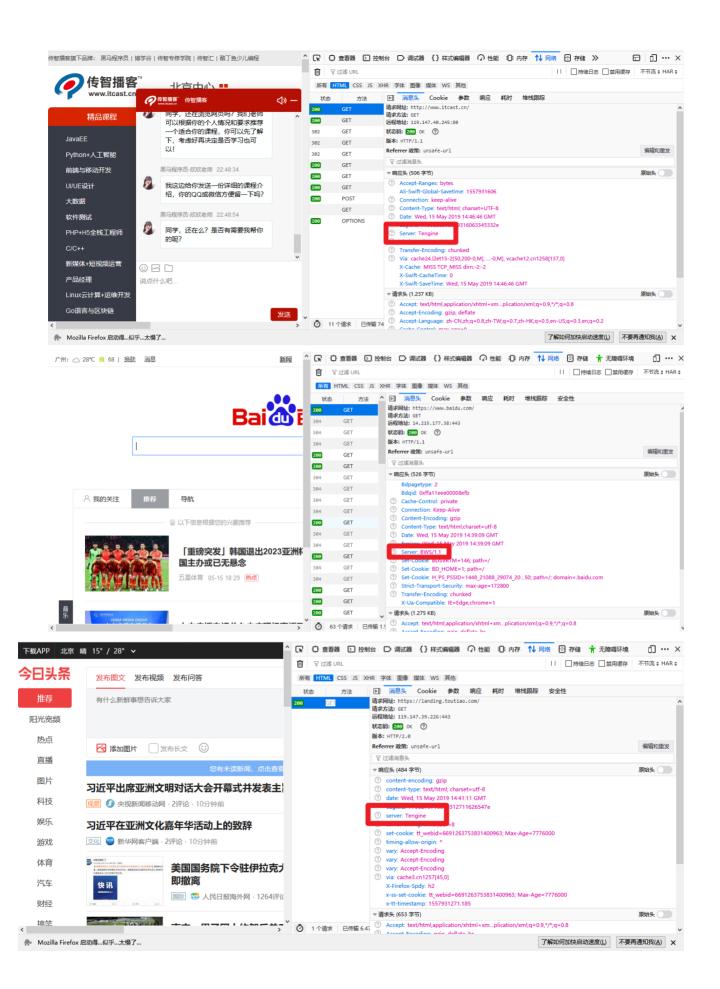
一、是什么

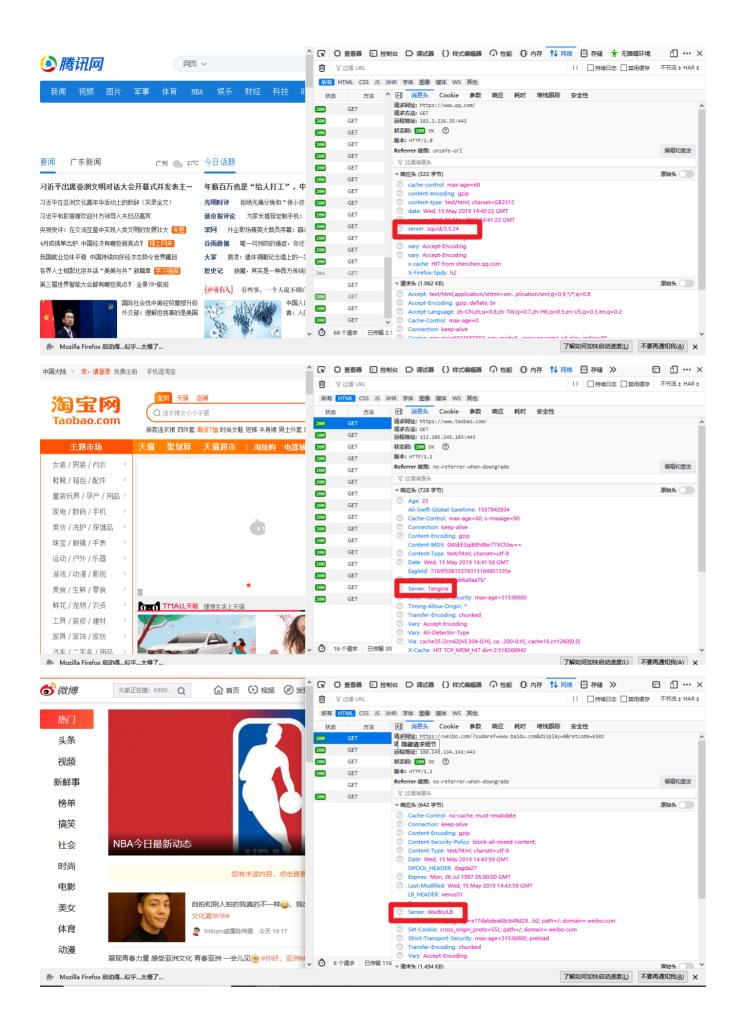
- 并行、并发:
 - 。 并发:
 - 你吃饭吃到一半,电话来了,你停了下来接了电话,接完后继续吃饭,这说明你支持并发。
 - 并行:
 - 你吃饭吃到一半,电话来了,你一边打电话一边吃饭,这说明你支持并行。
- 高并发 (High Concurrency)! = 并发
 - 是互联网分布式系统架构设计中必须考虑的因素之一;它通常是指,通过设计保证系统能够同时并行处理 很多请求。
- 高并发相关常用的一些指标
 - **响应时间**(**Response Time**): 系统对请求做出响应的时间。例如系统处理一个HTTP请求需要200ms, 这个200ms就是系统的响应时间。
 - **吞吐量 (Throughput)** : 单位时间内处理的请求数量。
 - **每秒查询率QPS (Query Per Second)** : 每秒响应请求数。在互联网领域,这个指标和吞吐量区分的没有这么明显。
 - **并发用户数**:同时承载正常使用系统功能的用户数量。例如一个即时通讯系统,同时在线量一定程度上代表了系统的并发用户数。

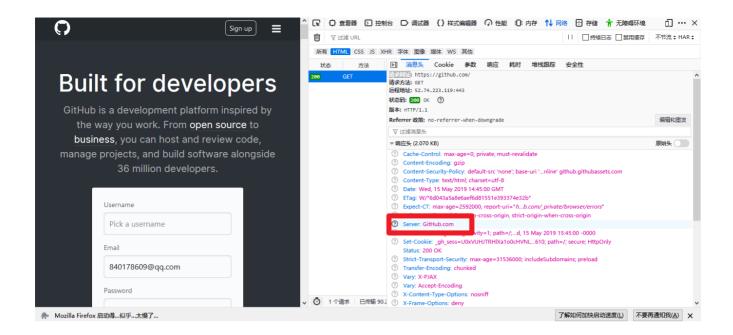
二、为什么

- 思考:
 - 为什么flask项目部署是基于ubuntu 16.04系统,使用 Gunicorn + Nginx 进行布署,云服务器为阿里云;而不是基于ubuntu 16.04系统,使用runserver进行部署,云服务器为阿里云?
- 例如: 小米手机每周二的秒杀,可能手机只有1万部,但瞬时进入的流量可能是几百几千万。
- 例如: 12306抢票, 票是有限的, 库存一份, 瞬时流量非常多, 都读相同的库存。读写冲突, 锁非常严重, 这是秒杀业务难的地方。那我们怎么优化秒杀业务的架构呢?

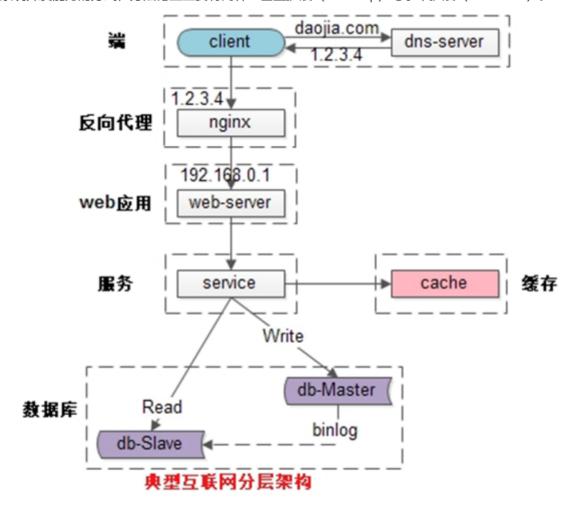
三、怎么做







- Tengine是由淘宝网发起的Web服务器项目。它在Nginx的基础上,针对大访问量网站的需求,添加了很多高级功能和特性。Tengine的性能和稳定性已经在大型的网站如<u>淘宝网,天猫商城</u>等得到了很好的检验。它的最终目标是打造一个高效、稳定、安全、易用的Web平台。
- 提高系统并发能力的方式,方法论上主要有两种:垂直扩展(Scale Up)与水平扩展(Scale Out)。



🗑 Django 本身提供了 runserver,为什么不能用来部署?

runserver 方法是调试 Django 时经常用到的运行方式,它使用 Django 自带的

WSGI Server 运行,主要在测试和开发中使用,并且 runserver 开启的方式也是单进程。

uWSGI 是一个 Web 服务器,它实现了 WSGI 协议、uwsgi、http 等协议。注意 uwsgi 是一种通信协议,而 uWSGI 是实现 uwsgi 协议和 WSGI 协议的 Web 服务器。uWSGI 具有超快的性能、低内存占用和多 app 管理等优点,并且搭配着 Nginx

就是一个生产环境了,能够将用户访问请求与应用 app 隔离开,实现真正的部署。相比来讲,支持的并发量更高,方便管理多进程,发挥多核的优势,提升性能。

高并发处理的思路及手段 水平扩容、垂直扩容的探讨 扩容 任务调度分布式elastic-job、 主备curator的实现、监控报 Redis, Memcache, Guava 警机制等 Cache等的介绍与使用 Kafka, RabbitMQ, 介绍切库、分表、支持 RocketMQ等队列特性介 高并发处理 多数据源的原理及实现 绍及使用队列的关注点 队列 思路与手段 服务降级的多种选择、 服务化.Dubbo与微服务 Hystrix介绍及使用等 Spring Cloud介绍 应用拆分 Guava RateLimiter的介绍与 使用、常用限流算法、自己

Cookie

实现分布式限流等

cookie 是一个非常具体的东西,指的就是浏览器里面能永久存储的一种数据,仅仅是浏览器实现的一种数据存储功能。

cookie由服务器生成,发送给浏览器,浏览器把cookie以kv形式保存到某个目录下的文本文件内,下一次请求同一网站时会把该cookie发送给服务器。由于cookie是存在客户端上的,所以浏览器加入了一些限制确保cookie不会被恶意使用,同时不会占据太多磁盘空间,所以每个域的cookie数量是有限的。

Session

session 从字面上讲,就是会话。这个就类似于你和一个人交谈,你怎么知道当前和你交谈的是张三而不是李四呢? 对方肯定有某种特征(长相等)表明他就是张三。

session 也是类似的道理,服务器要知道当前发请求给自己的是谁。为了做这种区分,服务器就要给每个客户端分配不同的"身份标识",然后客户端每次向服务器发请求的时候,都带上这个"身份标识",服务器就知道这个请求来自于谁了。至于客户端怎么保存这个"身份标识",可以有很多种方式,对于浏览器客户端,大家都默认采用 cookie 的方式。

服务器使用session把用户的信息临时保存在了服务器上,用户离开网站后session会被销毁。这种用户信息存储方式相对cookie来说更安全,可是session有一个缺陷:如果web服务器做了负载均衡,那么下一个操作请求到了另一台服务器的时候session会丢失。

Token

在Web领域基于Token的身份验证随处可见。在大多数使用Web API的互联网公司中,tokens 是多用户下处理认证的最佳方式。

以下几点特性会让你在程序中使用基于Token的身份验证

- 1.无状态、可扩展
- 2.支持移动设备
- 3.跨程序调用
- 4.安全

https://www.cnblogs.com/bobo-site/p/9317049.html