

# Nginx高并发网站技术架构实战



尚硅谷

讲师：张一明



# 亿级流量分布式系统架构设计

讲师：张一明



Nginx (engine x) 是一个高性能的HTTP和反向代理web服务器，同时也提供了IMAP/POP3/SMTP服务。Nginx是由伊戈尔·赛索耶夫为俄罗斯访问量第二的Rambler.ru站点（俄文：Рамблер）开发的，第一个公开版本0.1.0发布于2004年10月4日。

其将源代码以类BSD许可证的形式发布，因它的稳定性、丰富的功能集、简单的配置文件和低系统资源的消耗而闻名。2011年6月1日，nginx 1.0.4发布。

Nginx是一款轻量级的Web 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器，在BSD-like 协议下发行。其特点是占有内存少，并发能力强，事实上nginx的并发能力在同类型的网页服务器中表现较好，中国大陆使用nginx网站用户有：百度、京东、新浪、网易、腾讯、淘宝等。



## 通过扩容提升整体吞吐量

Nginx进阶系列课程中的内容不仅局限于Nginx自身功能讲解、配置、调优，我会由一个单一站点向高并发系统逐步演化，在演化的过程中以Nginx为核心系统化讲解企业级高并发技术解决方案



## 通过扩容提升整体吞吐量

### 扩容方式

1. 单机垂直扩容：硬件资源增加
2. 水平扩展：集群化
3. 细粒度拆分：分布式
  1. 数据分区
  2. 上游服务SOA化（原生支持水平/垂直扩容）
  3. 入口细分
    1. 浏览器
    2. 移动端原生App（物联网）
    3. H5内嵌式应用



## 通过扩容提升整体吞吐量

- 4. 数据异构化
  - 1. 多级缓存
    - 1. 客户端缓存
    - 2. CDN缓存
    - 3. 异地多活
    - 4. Nginx缓存
- 5. 服务异步化
  - 4. 拆分请求
  - 5. 消息中间件

### 扩容原则

- 1. 无状态原则
- 2. 弹性原则



## 单机垂直扩容：硬件资源增加

- 云服务资源增加
- 整机：IBM、浪潮、DELL、HP等
- CPU/主板：更新到主流
- 网卡：10G/40G网卡
- 磁盘：SAS(SCSI) HDD（机械）、HHD（混合）、SATA SSD、PCI-e SSD、MVMme SSD
  - SSD
    - 多副本机制
    - 系统盘/热点数据/数据库存储
  - HDD
    - 冷数据存储



## 机械磁盘







## 水平扩展：集群化

- 会话管理
  - Nginx高级负载均衡
    - ip\_hash
    - 其他Hash
      - hash \$cookie\_jsessionid;
      - hash \$request\_uri;
    - 使用lua逻辑定向分发
  - Redis + SpringSession



## Keepalive

- 在浏览器中查看是否启用keepalive
- 使用charles工具抓包监测
- 什么时候应该使用/关闭keepalive?
- 对客户端配置
- 对上游服务器配置



## AB 压力测试keepalive效果

- apache benchmark安装与基本使用
- 测试nginx直连
- 测试nginx代理
- 测试nginx Upstream keepalive 代理
- 测试Tomcat直连与代理
- 什么时候在Tomcat前置nginx 性能有明显提升?

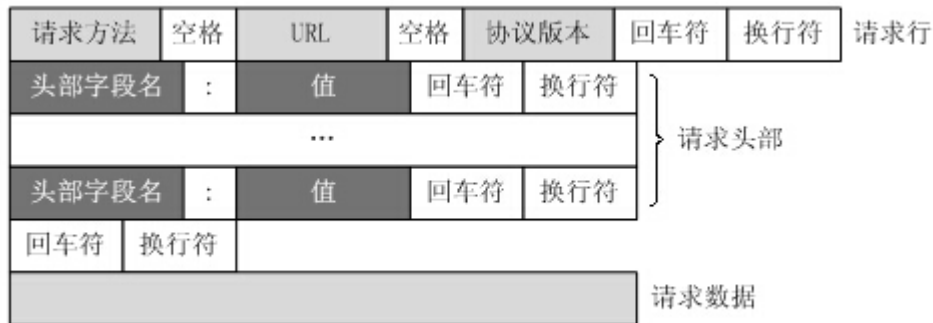


## Nginx反向代理核心流程

- Upstream proxy\_pass工作流程
- 内存与文件缓冲区
- 对客户端的限制



# Http1.x协议报文组成





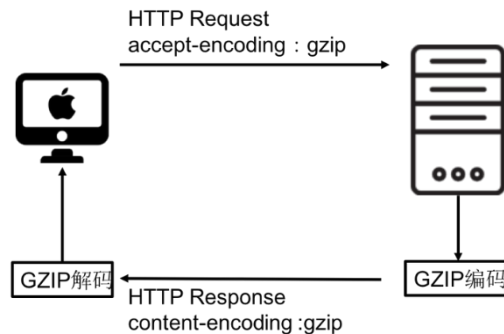
## 获得用户真实IP

- 上游服务器如何获得用户真实IP
- Http协议中有用的Header
- Nginx proxy\_set\_header



## Gzip压缩

- Gzip原理
- 使用gzip的站点
- Gzip动态压缩
- Gzip静态压缩
- 第三方zip模块 Brotli与模块化加载





## 合并请求

- 压缩请求数
- Concat模块原理及配置





## 资源静态化

- 高并发系统资源静态化方案
- 一致性问题
- 合并文件输出
- 集群文件同步



## 多级缓存

- 静态资源缓存
- 浏览器缓存
- CDN缓存
- 正向代理缓存
- 反向代理缓存
- Nginx内存缓存
- 外置内存缓存
- 上游服务器应用缓存



# 缓存文件元数据信息提高sendfile性能

- Strace内核执行过程追踪神器
- Nginx发送文件过程
- open\_file\_cache配置优化



# Nginx外置缓存

- error\_page
- 匿名location
- Memcached使用
- Redis使用



## 第一部分：扩容



### 浏览器缓存

- 多级集群负载时last-modified必须保持一致
- 还有一些场景下我们希望禁用浏览器缓存。比如轮训api上报数据数据
- 浏览器缓存很难彻底禁用，大家的做法是加版本号，随机数等方法。
- 只缓存200响应头的的数据，像3XX这类跳转的页面不需要缓存。
- 对于js, css这类可以缓存很久的数据，可以通过加版本号的方式更新内容
- 不需要强一致性的数据，可以缓存几秒
- 异步加载的接口数据，可以使用ETag来校验。
- 在服务器添加Server头，有利于排查错误
- 分为手机APP和Client以及是否遵循http协议
- 在没有联网的状态下可以展示数据
- 避免流量消耗过多
- 提前下发 避免秒杀时同时下发数据造成流量短时间暴增
- 兜底数据 在服务器崩溃和网络不可用的时候展示
- 临时缓存 退出即清理
- 固定缓存 展示框架这种，可能很长时间不会更新，可用随客户端下发
- \*\*首页\*\*有的时候可以看做是框架 应该禁用缓存，以保证加载的资源都是最新的
- 父子连接 页面跳转时有一部分内容不需要重新加载，可用从父菜单带过来
- 预加载 某些逻辑可用判定用户接下来的操作，那么可用异步加载那些资源
- -漂亮的加载过程 异步加载 先展示框架，然后异步加载内容，避免主线程阻塞



## CDN缓存

- 内容分发网络
- 进一步向用户逼近

### 职责描述:

- 1、负责 CDN系统设计和研发工作;
- 2、负责 NGINX/OpenResty 相关开发工作。

### 任职要求:

- 1、本科以上学历，3年以上开发经验。
- 2、具有工作热情和责任感，能承担工作压力。
- 3、有基本的数据结构与算法功底;
- 4、扎实的网络层面相关问题解决能力;
- 5、熟悉 Linux 平台编程和网络编程;
- 6、熟悉 DNS/HTTP/HTTPS/QUIC 其中一种协议及部分实现;
- 7、熟练掌握其中一门编程语言 Python/Golang/Lua/Rust/C;
- 8、熟悉服务端主流高性能开源组件一些设计细节，例如NGINX/OpenResty/Redis

\* 加分项：有内核层面 IO 网络及相关调优经验。

\* 加分项：有 NGINX/OpenResty 和 CDN 相关开发经验。

\* 加分项：熟悉一些性能调优工具，例如 Perf/SystemTap/FlameGraph。

\* 加分项：熟悉 Kafka / ES 等大数据分析处理相关开源组件



## 地理位置

- GEOIP2

Dear zhangyiming,

Thank you for creating an account.

Your username is [REDACTED]. To log in, you must first create a password using the following link: <https://www.maxmind.com/en/account/set-password?token=EBCFC3050CFB4F33835189E2E5BCC04D>

This password link will expire in seven days.

1. [Access your MaxMind account.](#)
2. [Learn how to manage your MaxMind Account on our knowledge base.](#)
3. [View our developer's documentation and APIs.](#)

Thanks again for choosing MaxMind!

Sincerely,  
The Team at MaxMind

---

minFraud service customers can help us detect and prevent more fraud by reporting chargebacks, false positives, and suspected fraud [here](#).

Follow us on [Twitter](#).

Connect with us on [LinkedIn](#).

For information about MaxMind's privacy practices, see our [Privacy Policy](#).



## Nginx Stream代理

- Mysql透明化多主高可用，负载均衡配置





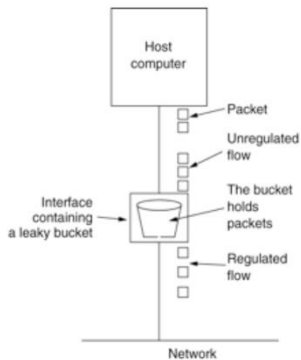
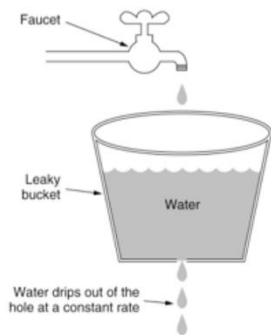
## 水平扩展：集群化

- 云服务资源增加
- 整机：IBM、浪潮、DELL、HP等
- CPU/主板：更新到主流
- 网卡：10G/40G网卡
- 磁盘：SAS(SCSI) HDD（机械）、HHD（混合）、SATA SSD、PCI-e SSD、MVMme SSD
  - SSD
    - 多副本机制
    - 系统盘/热点数据/数据库存储
  - HDD
    - 冷数据存储



## Nginx qps限制

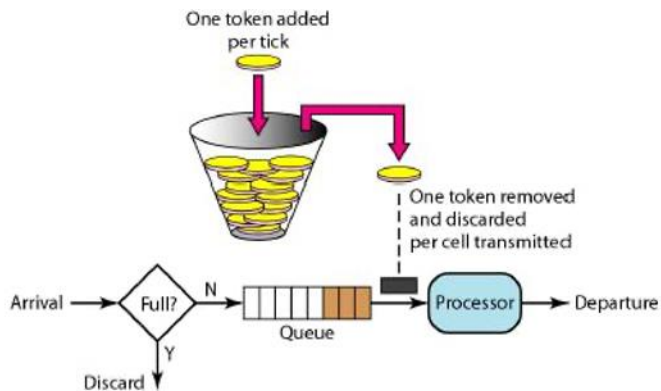
- 漏桶算法与令牌桶算法
- limit\_req





# Nginx 带宽限制

- 漏桶算法与令牌桶算法
- `limit_req` `limit_rate`





# Nginx 并发数限制

- 计数器算法



# 使用Lua对Nginx二次开发

- Lua luajit nginx openresty关系
- Lua开发环境
- Lua基础语法



# Openresty

- 安装
- 测试lua脚本
- 获取系统变量



# Openresty常用模块

- Nginx进程缓存
- 连接redis
- 连接mysql
- 模板引擎
- 基于Openresty的开源项目



谢谢观看