ngnix可以抗住5w的并发，qps每秒查询率

tomcat可以抗住200以下的并发

第一步可以通过ngnix进行削峰

最前端某一个业务线，业务线已经被细分了，单业务线下也拥有极高并发，物理结构上可以划分域名来进行拆分，一个域名对应一组业务线。

域名

CDN，缓存静态资源如：静态图片和html和少量一致性低的动态缓存，更新动态缓存，使用JOB：定时更新任务。

LVS

反向代理ngnix：单纯作为负载均衡使用

应用层ngnix：ngnix + lua + redis，ngnix + lua + mq ngnix加上lua可以在后端之前解决一部分访问压力

redis缓存，任何接口，sql语句，都可以当过redis的key，拿到的数据当做值，缓存解决高并发

浏览器找域名顺序🡪hosts🡪本地dns缓存🡪dns服务器地址 ttl定时更新本地dns

高并发考虑因素，并发量的级别

一致性的要求

硬件层面的负载均衡f5集群和集群间负载

软件层面的负载均衡LVS,Ngnix主机和主机间的负载均衡

dns解析技术

防火墙技术

ngnix

redis集群

mysql主从，读写分离，分库分表

EKK日志收集

mq kafka集群

etcd数据库，优点占用资源非常小 1w一下并发只需要2核8g就可以

coreOs操作系统，优点大小非常小，作为容器启动非常快

分布式锁，总线协议，管城

watch dog

redis是单线程的

redis执行lua脚本会整个执行完才返回，一次只能有一个线程能进来并执行脚本

红锁，一台服务器必须拿到n/2+1台redis的锁才算作抢到锁

不能用哨兵，是因为哨兵模式，切换服务也会造成锁丢失

单实例，锁死了，报警，退化成为另一个锁。

悲观锁和乐观锁

平时先走redis悲观锁，在走mysql。如果redis挂了，就走mysql，并进行限流

客户端循环对redis加锁，超过n/2+1个成功时，对加锁成功的redis并行发出刷新加锁时间的到10s的命令，如果最终加锁失败则对所有的加锁成功的redis执行del删除锁操作

守护线程的续租

Martin Kleppmann文章，对红锁进行质疑的一篇文章，在redis官网中。