# 如何应对高并发？

高并发要细化，那么则分为前端、中端、后端，我相信大家都会很疑惑为什么会有前中后不同的端，当然每一个人对高并发以及技术、架构等理解不一样，所以每一个人的定义也就不一样，毕竟这些都是理论上定义，只要是在实际业务场景中应用是正确的无论如何定义都是对的。

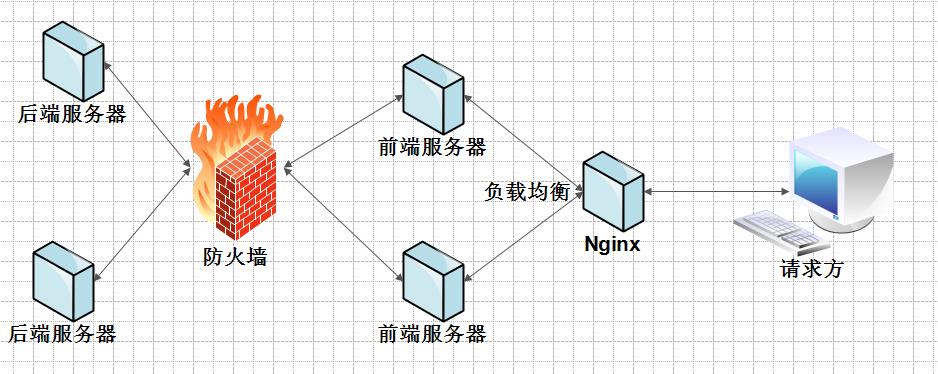
高并发的核心在于集群，通过集群提高处理能力，以下的都是基于集群的架构。

## 前端

前端主要就是对于页面的访问处理，目前一般的解决方案是使用Nginx，因为F5价格太贵了，所以这里就不多阐述。

Nginx可以通过前端的请求转发、对后端服务器进行代理、面对并发请求时，且是集群架构时进行负载均衡。

案例：前端页面使用HTML+CSS进行渲染，并使用Node.js进行编写简单的业务逻辑，如果有复杂的业务逻辑则请求后端服务器进行处理，那么前端则只需要关注页面的渲染，Nginx则只需要关注请求包的转发，对请求进行负载均衡的处理以及返回前端页面给请求方。

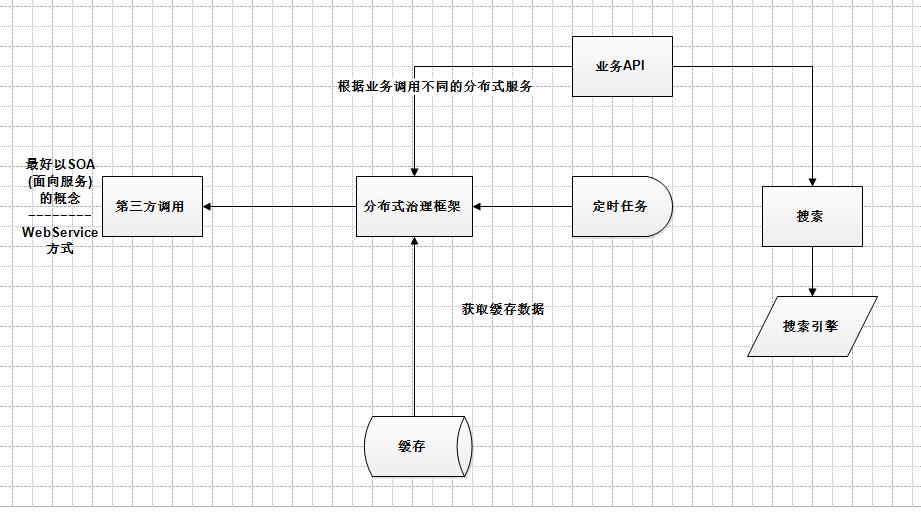


## 中端

我相信大家都会很疑惑，中端是指什么？

中端指的就是中间的那一层，也就是业务逻辑处理那一层，在面对大量的高并发请求处理，中端一般对常用的业务数据进行缓存，在缓存完了之后对大量的请求进行多线程分批处理。

中端的业务处理还包括了操作数据等一系列操作，这里的解决方案有很多，可以通过中间件、或Mongodb+Spark的SparkSQL组件来提高性能等……这些百度上一搜索就有很多就不多阐述了。



## 后端

这里所说的后端就是对于数据库的操作，在中端的数据时候就已经涉及到了大量的数据操作，但是面对如此之多的并发，数据库的性能会剧烈的下降，那么如何保证数据库的性能？  
或者换个思路，如何快速获取到业务所需要的数据，或者存储业务产生的数据？

分库分表，这些老套路就不多说了。

找专人做专事，如ES搜索引擎是在搜索的时候是很快的，那么如果把ES当成一个缓存数据进行查找数据行不行？  
 如Redis是一个缓存数据库，那么使用Redis存储数据，或者开多一个线程存储数据，或者使用Mongodb存储数据，在设定一个定时器到一定时间就进行同步到MySQL数据库中。

但是这里就涉及到一个同步问题，这个问题请持续关注此博客。

