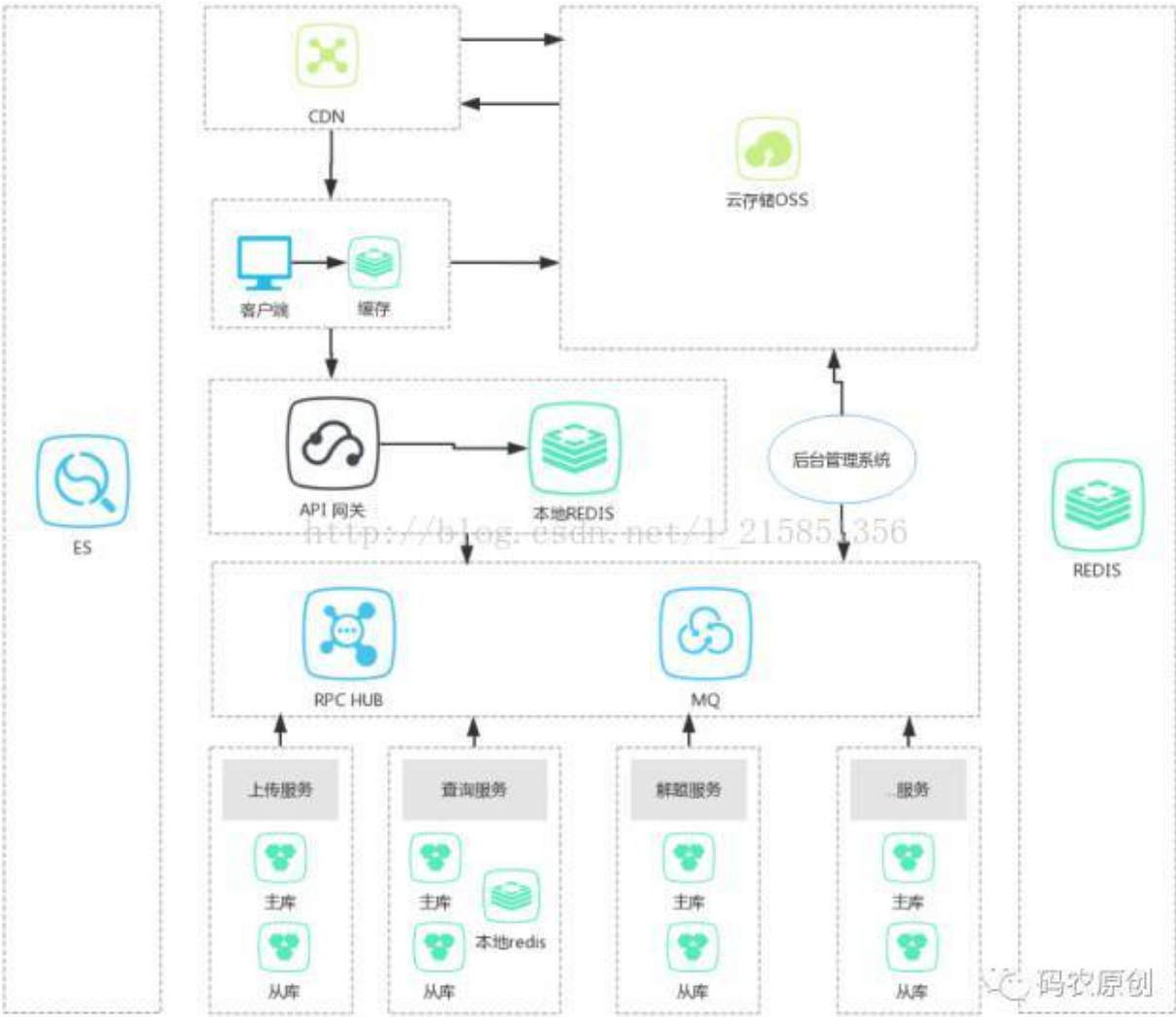


这个模式可以说是应对超高查询压力的一种普遍采用的策略，基本的思想就是在所有链路的地方，能加缓存就加缓存，如下图所示：



如上图所示，一般在三个地方加入缓存，一个是客户端处，一个是API网关处，一个是具体的后端业务处，下面分别介绍。

客户端处缓存：这个地方加缓存可以说是效果最好的——无延迟。因为不用经过长长的网络链条去后端业务处获取数据，从而导致加载时间过长，客户流失等损失。虽然有CDN的支持，但是从客户端到CDN还是有网络延迟的，虽然不大。具体的技术依据不同的客户端而定，对于WEB来讲，有浏览器本地缓存、Cookie、Storage、缓存策略等技术；对于APP来讲，有本地数据库、本地文件、本地内存、进程内缓存支持。以上提到的各种技术有兴趣的同学可以继续展开来学\*\*。如果客户端缓存没有命中，那么就会去后端业务拿数据，一般来讲，都会有个API网关，在这里加缓存也是非常有必要的。

API网关处缓存：这个地方加缓存的好处是不用把请求发送到后方，直接在这里就处理了，然后返回给请求者。常见的技术，如http请求，API网关用的基本都是nginx，可以使用nginx本身的缓存模块，也可以使用Lua+Redis技术定制化。其他的也都大同小异。

后端业务处：这个我想就不用多说了，大家应该差不多都知道，什么Redis, Memcache, Jvm内等等，不熬述了。

实践中，要结合具体的实际情况，综合利用各级缓存技术，使得各种请求最大程度的在到达后端业务之前就被解决掉，从而减少后端服务压力、减少占用带宽、增强用户体验。至于是否只有这三个地方加缓存，我觉得要活学活用，心法比剑法重要！总结一下这个模式的优缺点：

优点：抗住大量读请求，减少后端压力。

缺点：数据一致性问题较突出，容易发生**雪崩**，即：如果客户端缓存失效、API网关缓存失效，那么所有的大量请求瞬间压向后端业务系统，后果可想而知。