Scenario 场景： 注册，登录，查询，用户信息修改等

假设支持100M DAU，即1亿日活跃用户

注册，登录，信息修改QPS（每秒查询率）

100M \* 0.1/86400 大约为100

0.1代表平均每个用户每天登录+注册+信息修改次数

峰值=100\*3=300

查询的QPS：

100M \* 100/86400，大约为100k

100代表平均每个用户每天与查询用户信息相关的操作（查询好友，发信息，更新消息主页等）

峰值=100k\*3=300k

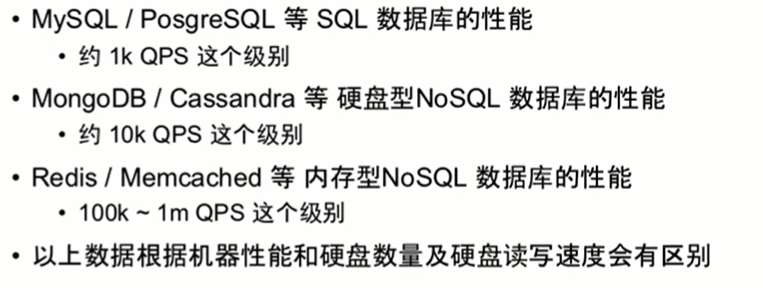
Service 服务：

一个AuthenticationService负责登录注册（如果DAU很大，也可以把登录和注册分成两个系统），一次登录后后续不用再重新输入用户名/密码

一个UserService负责用户信息存储与查询

一个FriendshipService负责好友关系存储

分析QPS的原因：QPS的大小决定了数据存储系统的选择。



用户系统的特点：读非常多，写非常少。一个读多写少的系统一定要使用Cache进行优化。

缓存：把之后可能要查询的东西先存一下，下次要的时候，直接从这里拿，无需重新计算和存取数据库等。

Cache类似一个哈希表结构，key-value

常用的Cache软件：

Memcached(不支持数据持久化)

Redis(支持数据持久化)

Cache不一定存在内存中，尽管绝大多数情况下存在内存中。

文件系统也可以作为Cache，例如网络请求的缓存，计算量很大的工作结果的缓存，这些操作的访问速度都比文件系统慢。文件系统一般不作为数据库的缓存，二者的访问速度接近。

CPU也有Cache，L1 Cache和L2 Cache，这两个缓存大，电脑运行速度快。

Cache与哈希表的区别：

1. Cache是一个软件，一个系统，而哈希表是数据结构。Cache除了提供哈希表的功能外，还会提供一套协议，让其它机器能访问到这台机器的Cache
2. Cache会有一个淘汰机制，著名的LRU算法。
3. Cache有个特别的操作，可以设置超时时间，让某个key-value只存在一定的时间。