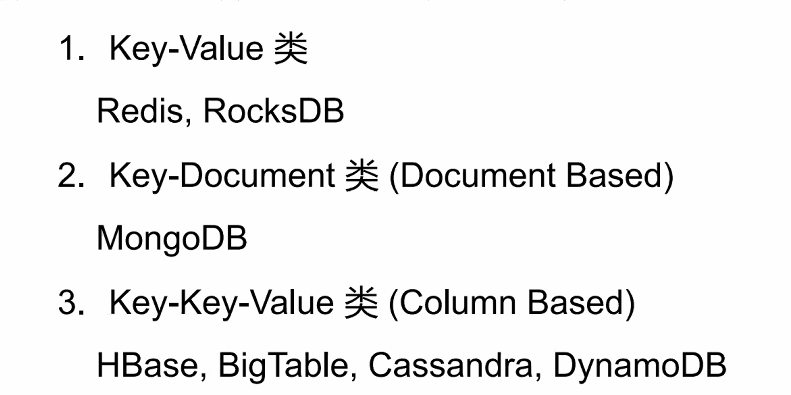
nosql：硬盘上的哈希表，常见的有mongoDB，Redis，HBase，RocksDB。

HBase访问时不需要用户名和密码，所以一般情况下都配置成某类或某些机器内部网络访问，不对外提供。



MongoDB中 的Document其实是一个json格式的哈希表。

Key-Key-Value其实就是哈希表套哈希表，要想访问数据，首先要拿到第一个Key，row key，再拿到第二个Key，column key，才能最终获取到数据。

nosql最初就是为了解决日志的大量日志，尽可能提高写的效率。

对于读少写多的情况，读的时候稍微慢点无所谓，写要尽可能快。

nosql，key-value的数据库，相比mysql，结构简单，qps比mysql至少是10倍大小。

选择HBase：

1. 高并发
2. 最主要就是天然的分布式功能，带来了安全，可靠的提升。
3. 写优先，写多读少。

HBase是一个在硬盘上的大哈希表，外层是row key，它对应的value又是一个哈希表，其对应的key是column key。如果去对应java的数据结构，就是HashMap<string, HashMap<string, string>>，对应C++就是unordered\_map<string, unordered\_map<string, string>>。其中row，column和mysql中的行，列不太一样，它是一级维度，二级维度的意思。

HBase是基于row key的全局排序，这意味着不同机器上存储的key也是有序的。column key是在每个row 可以下面，它是无序的。

对row key可以进行range query，范围查询，例如指定row key在’abc’到’def’之间的所有数据。

HBase是java写的，所以需要jdk。

HBase ORM的设计：直接调用happybase的接口会比较麻烦，包括happybase接口自身的参数在程序里直接传递比较麻烦，另外HBase的row key的设计也需要技巧，这样就需要对进行一层封装，构建一个类似Django ORM的HBase ORM，这样调用hbase时能更加方便，另外希望包装的接口更类似Django ORM，方便代码风格统一。

异步任务的重试：第三方的API都不可信，需要重试。

column\_family：可以把一部分column key进行组合，通常是把需要一起访问的column key放到一个column family中，这样hbase会把他们存在一起，提高读取速度。