DB Proxy

1. 动态路由（扩展性）
   1. 一致性hash
      1. 可参考redis slot机制
      2. 余数hash，一致性hash［0-（232-1）］，二叉查找树（查找不小于查找数的最小值，最右边的叶子节点与最左边的叶子节点相连，构成环），一致性hash的数据平衡（负载不均衡），一台物理设备对应150个虚拟节点。
      3. 不能命中率（n/（n+x））。n当前设备数，x新添加的设备数
      4. 命中率1-（x／（n＋x））。同上
   2. 支持软负载，多路由节点协同工作
      1. array/f5／haproxy，直接将消息转发至路由集群（一期优先采用的方案）。
      2. DB Proxy Gateway，转发消息至路由集群。Gateway需要支持主备和负载（二期考虑自实现，以降低成本）。
   3. 支持高可用（路由器支持主从机制）
      1. 如果负载方式采用array/f5/haproxy的方案，路由器可以不支持主从机制，可通过增加路由节点的方式实现高可用。弊端？
   4. 微内核，插件方式扩展
      1. 通信模块
      2. mem/redis协议转码模块
      3. hbase/mysql/es持久化模块
      4. 模块间的通信机制，url + netty,可参考dubbo URL配置实现
      5. 支持failover机制
      6. web控制台
      7. 可通过控制台动态扩容
      8. 路由机制：最少使用，轮询，加权（处理能力，设备配置等）
   5. 支持json,memcached和redis协议
   6. 支持mem/redis数据直接存储，以及mem/redis数据持久化至hbase,mysql,es。Mysql（mysql binlog）持久化至mem/redis。存储引擎可用插件方式扩展。
2. 动态扩容（伸缩性）
   1. 路由器中的一致性hash应该属于动态扩容的基础。
   2. 动态扩容除一致性hash以外，主要体现的是数据动态迁移，数据一致性（要区分缓存服务（允许少量数据的冗余和丢失，qps优先）和持久性服务的特性（持久性，正确性和可用性，在任何时候都必须可用））。根据应用的不同状态，以及存储容器的状态自行伸缩的一种体现。
   3. 一期，可通过控制台人工干预，根据应用的数据量，目标存储融资的性能进行扩容。
   4. 二期，通过动态监控目标应用和存储容器的数据量和性能，通知控制台自动部署存储容器（mem/redis/mysql）。需要云支持。