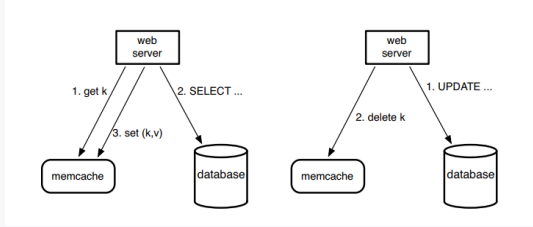


Memcached

意图

- 提供可靠的、大规模的缓存
- memcache 架构简单
- memcache 性能的优化
 - 通信优化，使用 UDP，实现 Get 语义
 - 减轻 db 的压力
 - 过期集合 (stale sets) 和 thundering herds, 实现了租约
 - 使用 memcache pool 和 pool 内部的副本 提供tidings层次的共享
 - memcache failure 导致的 cascading failure (雪崩) 引入 Gutter
- 在单 Region 引入 replication
- 跨地区的主从备份中保持一致性
- 单机优化



结构

- memcache 缓存系统
- memcached 旁路缓存的服务器
- 缓存结构
 - look-aside cache (旁路缓存) 缓存不与微服务和数据库进行串联
 - inline cache cache 在 微服务和数据库之间串联
- look-aside cache
 - Get 操作 read from cache, 先读缓存
 - Set 操作 read from db, 未命中, 从数据库中读, 在更新cache
 - update db 然后在验证缓存是否有效 (invalid cache)
- memcached 客户端
- proxy 服务器 用于 batch 和 proxy
- 结构存在的问题
 - 可能会有热数据和陈旧的/很少更新的数据
 - 复制是必要的, 但是可能会导致 fan-out (复制的分支) 很大, 即更新一处需要更新很多的副本或者 cascade 的关系
 - 读 >>> 写, 作为 cache, 读是主要的负载

集群中的延迟 (Latency) 和负载 (Load)

- 延迟
 - 作为 web 应用的通用缓存, Get 的模式于读取数据库的 KV 不同, 而是在 Get 一个 网页的时候, 而会获取更多的子组件的数据, 甚至相互之间有加载的依赖关系, 会给服务器造成严重的负载
 - 代理服务器 (proxy)
 - memcached 客户端
 - 提高性能的方法
 - 客户端在读请求使用 UDP + 拥塞控制实现 降低 TCP 连接带来的 CPU 和 内存开销
 - 实现了慢启动 + 滑动窗口
- 负载
 - stale sets (过期集合) 给数据库带来的压力 使用 lease (64bit 随机 token) 保证写版本正确
 - Thundering herds 同样是 lease, 限定 lease 的频率

灾难恢复

- 缓存的 code
 - 指代
 - 可能挂掉重启, 内存全丢失
 - 可能刚拉起来, 内存没有东西
 - 策略
 - 直接从别的 warm 集群 cache 读, 优化效率
- 跨集群读会造成的一致性问题解决
 - delete 有 hold-off 时间
 - hold-off 中, 对这个 key 添加会失败
 - 重新读 db

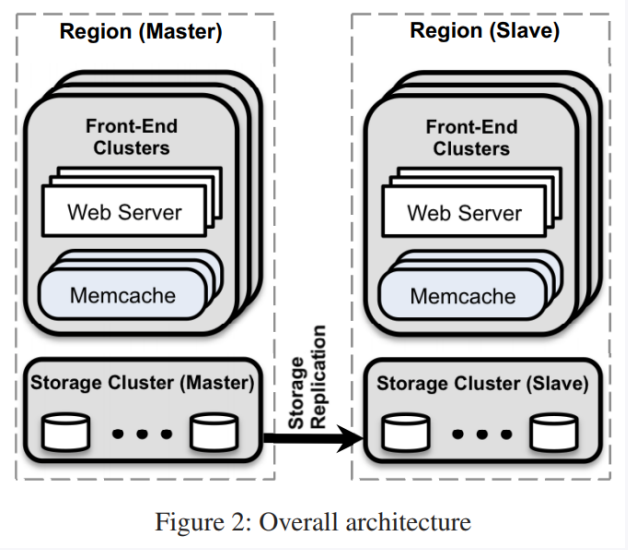
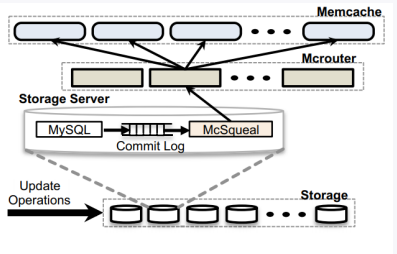


Figure 2: Overall architecture

全部的结构



Region

- 一组 frontend cluster, 多组 存储到底层 storage 上
- 代理服务器, 控制数据访问, 问 memcache 降低负载
- Mcrouter
- Replication 同一机器实现不同的 Region, 来保存不同的数据
- replica 和 冷热分割

handling Failures

- 造成雪崩的原因: 池子掉线, 数据打到数据库中
- pool 掉线不会对数据库造成影响, 因为是 cache
- 故障解决
 - 由于网络或服务故障而导致无法访问少量主机
 - 自动修复
 - 使用 空闲节点 (Gutter) 来接管故障服务器的职责
 - 影响集群中很大比例的服务器广泛停机

memcache pool

- 按照 key 分割 memcache
 - 各个应用访问模式不同, 会有不同读写频率。混合负载会导致负面干扰
 - 把 冷热-高低代价 的数据放在不同的 pool
- 通过 pool 内部的 replication 降低负载