

课时35

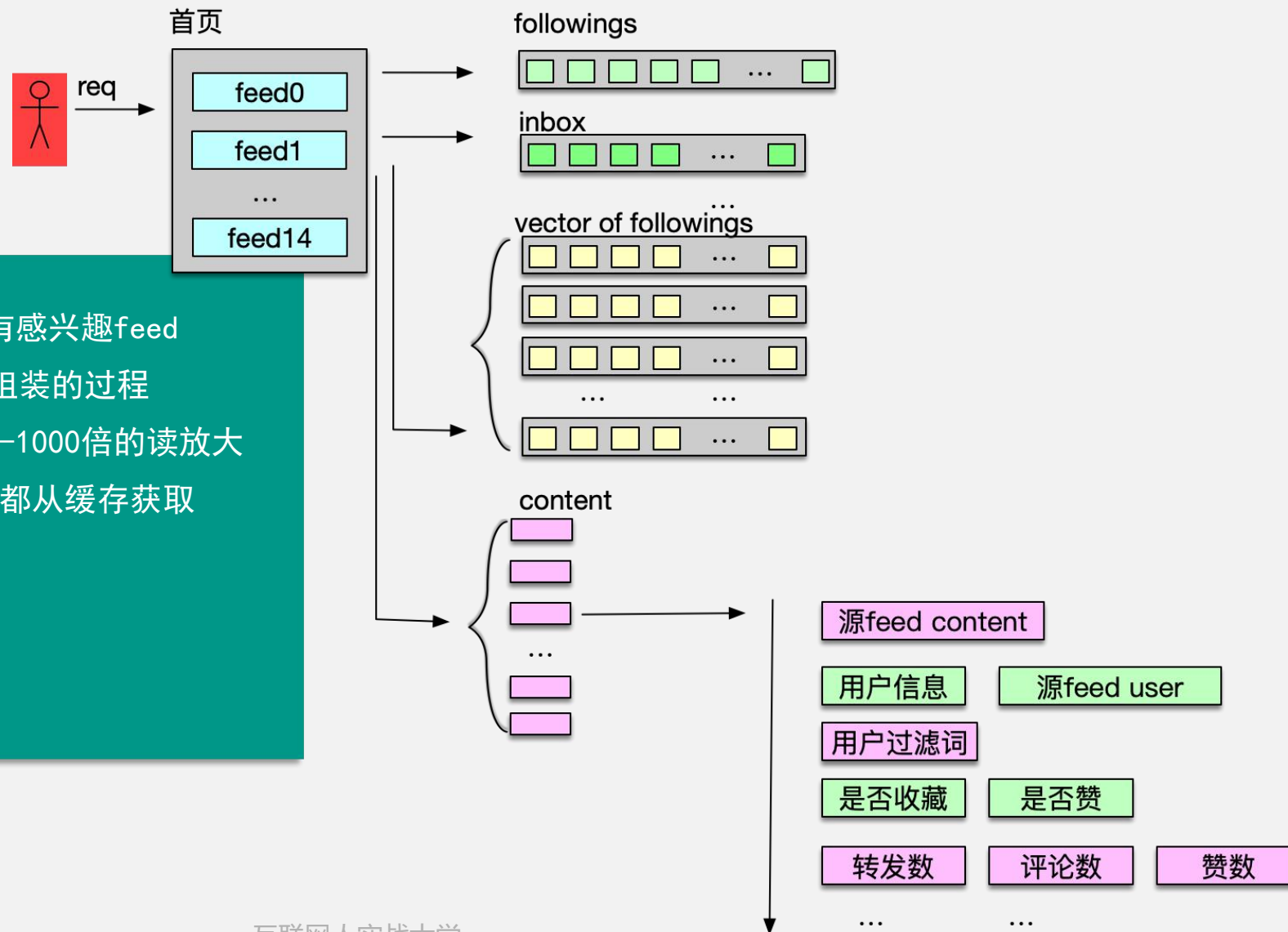
如何为社交feed场景设计缓存体系？

1. Feed 流场景分析
2. Feed 流流程分析
3. Feed 流缓存架构
4. Feed 流 Mc 架构

社交Feed流缓存解析

Feed流 场景分析

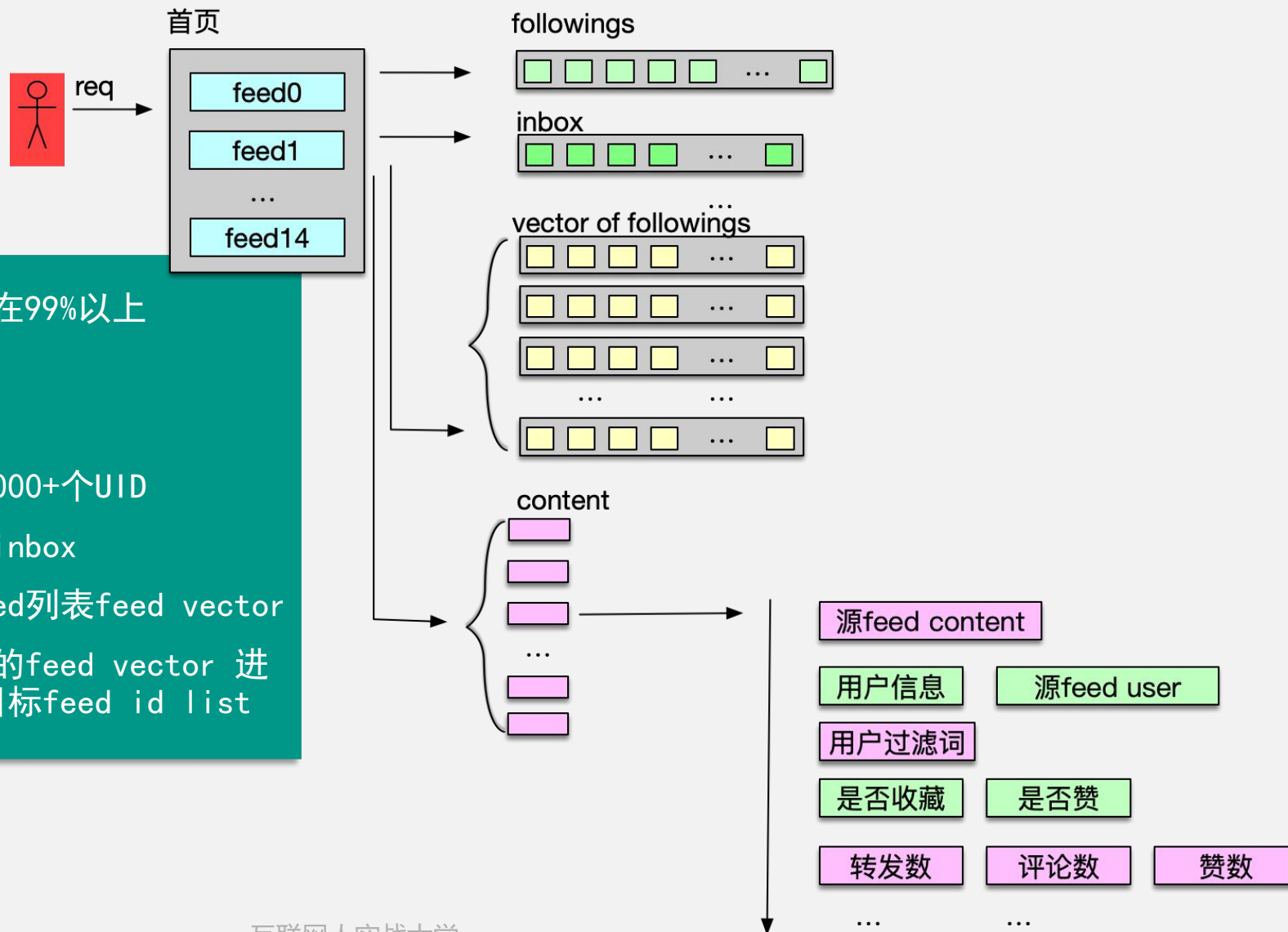
- Feed 流获取，是一个获取所有感兴趣feed并对feed进行过滤、动态组装的过程
- Feed 流请求的处理，存在100-1000倍的读放大
- 重度依赖缓存，所有数据基本都从缓存获取
 - 关系：Redis
 - Vector：Memcached
 - 计数：计数服务



社交Feed流缓存解析

Feed流 流程分析

- 核心业务，缓存命中率要求在99%以上
- 需要多线程并发获取及组装
- Feed 处理流程
 - 获取关注关系，300-2000+个UID
 - 获取用户自己的feed inbox
 - 获取所有关注UID的feed列表feed vector
 - 将inbox 和 关注用户的feed vector 进行排序、分页，获得目标feed id list

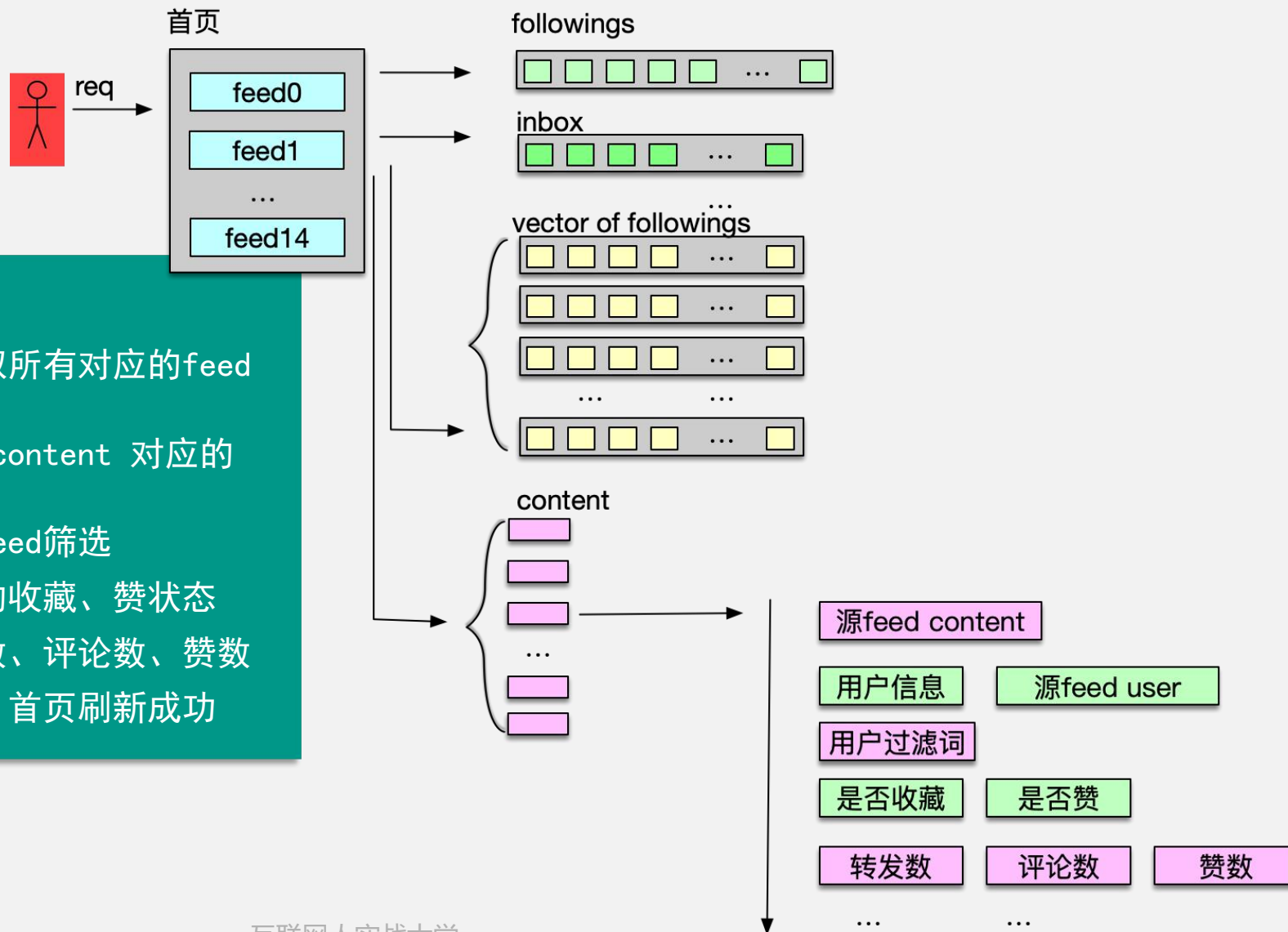


社交Feed流缓存解析

Feed流 流程分析

• Feed 处理流程

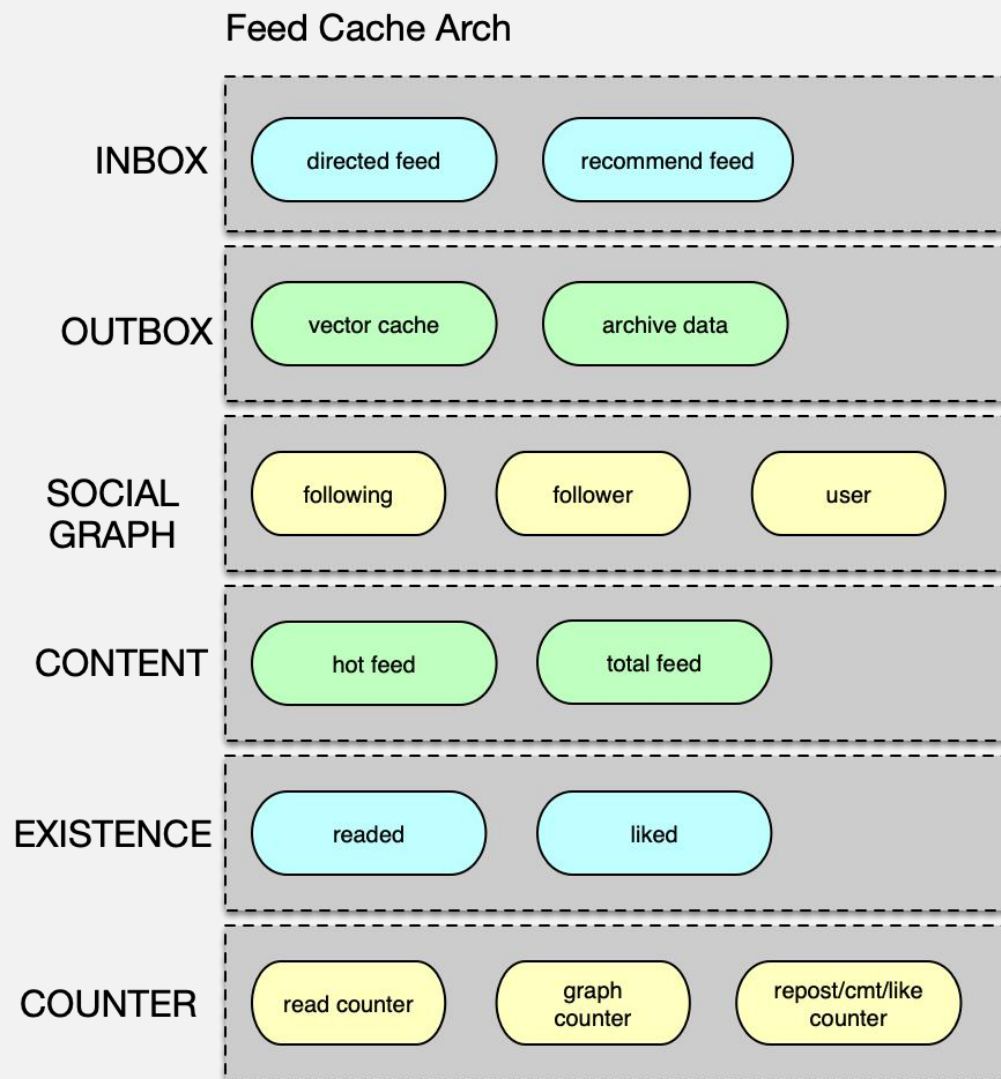
- 根据feed id 列表获取所有对应的feed content
- 获取并组装每条feed content 对应的user、parent-feed
- 根据用户过滤词进行feed筛选
- 获取用户对这些feed的收藏、赞状态
- 获取这些feed的转发数、评论数、赞数
- 组装完毕，返回用户，首页刷新成功



社交Feed流缓存解析

Feed流 缓存架构

- 缓存业务数据主要分为6大类
 - Inbox 收件箱，部分可见的特殊feed
 - Outbox 发件箱，所有人可见的feed
 - Socail Graph 关系，关注、粉丝列表
 - Content feed 内容
 - Existence 存在性判断，是否阅读，是否赞
 - Counter 计数服务，阅读数，关注数，粉丝数，feed 的转发、评论、赞数

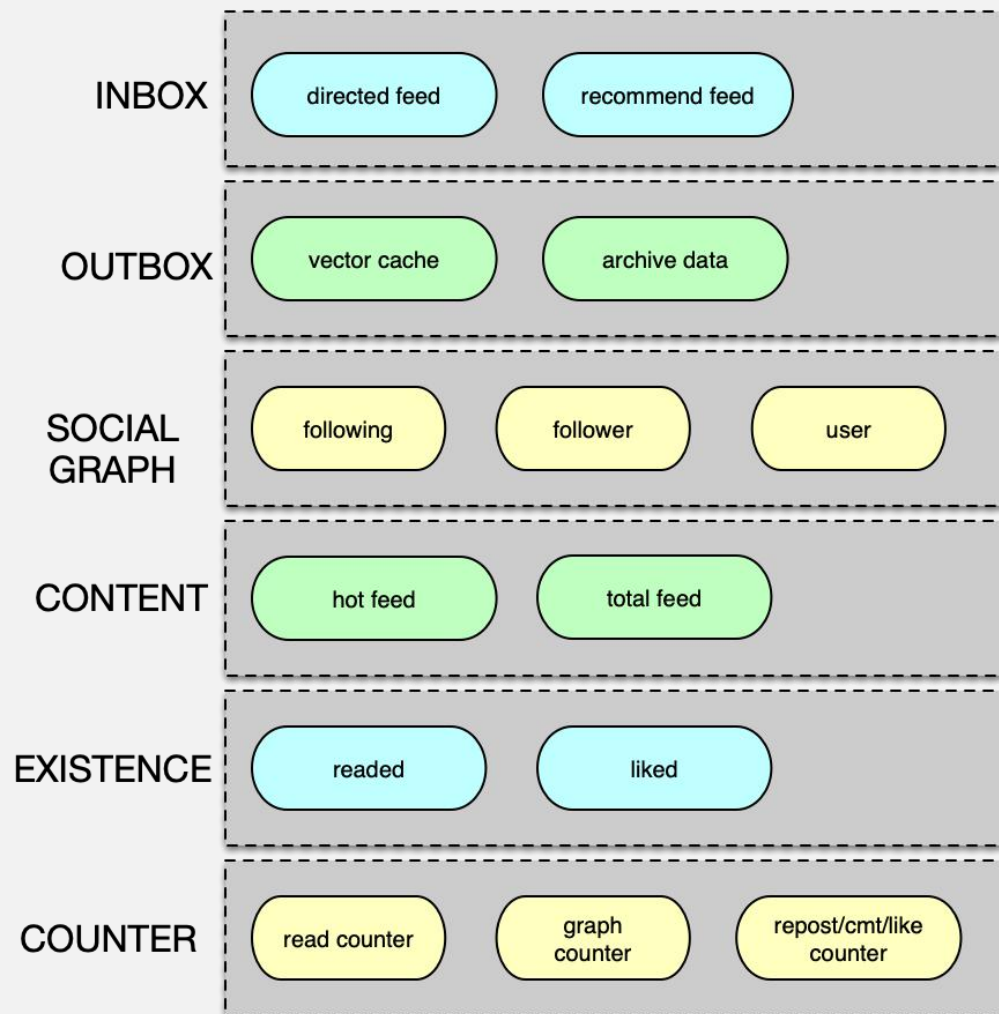


社交Feed流缓存解析

Feed流 缓存架构

- Feed 收件箱、Feed 发件箱
 - 缓存类型: Memcached
 - 数据格式: id/对象 一维数组
- 关注列表
 - 缓存类型: Redis
 - 数据格式: longset
- Feed Content
 - 缓存类型: Memcached
 - 数据格式: Google protocol buffers

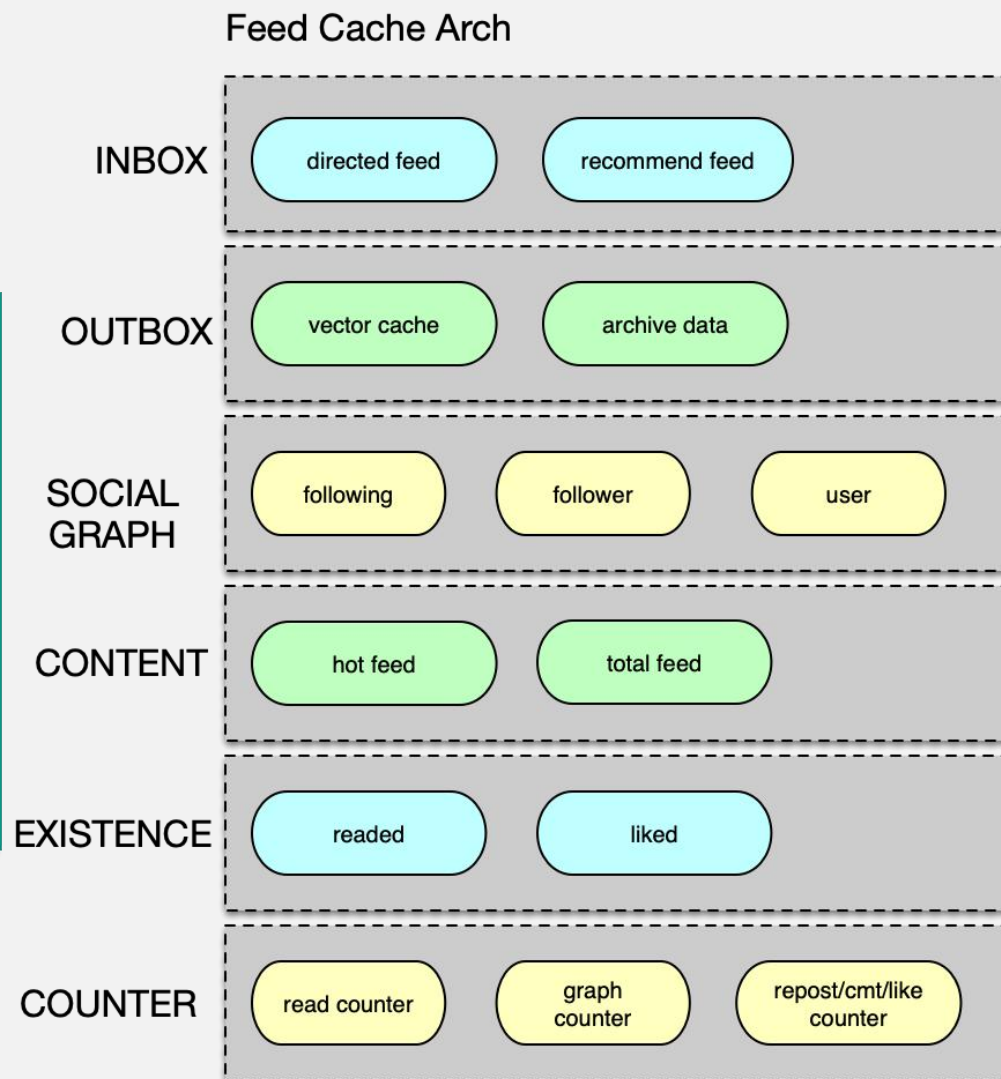
Feed Cache Arch



社交Feed流缓存解析

Feed流 缓存架构

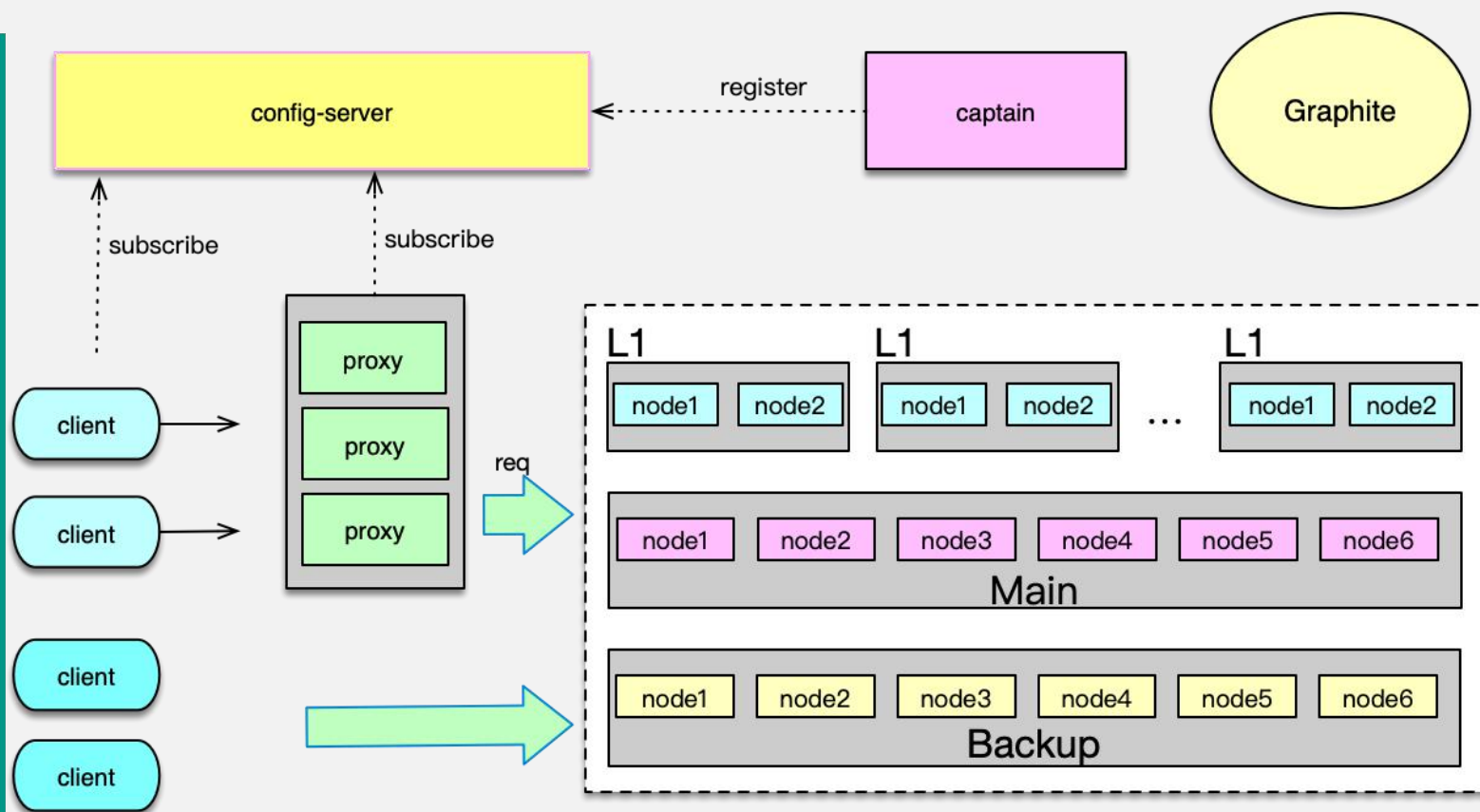
- 存在性判断
 - 缓存类型: Phantom
 - 数据格式: Bloom Filter
- 计数服务
 - 存储类型: CounterService
 - 数据格式: schema, id-val0+val1+val2+val3



社交Feed流缓存解析

Feed流 Mc 架构

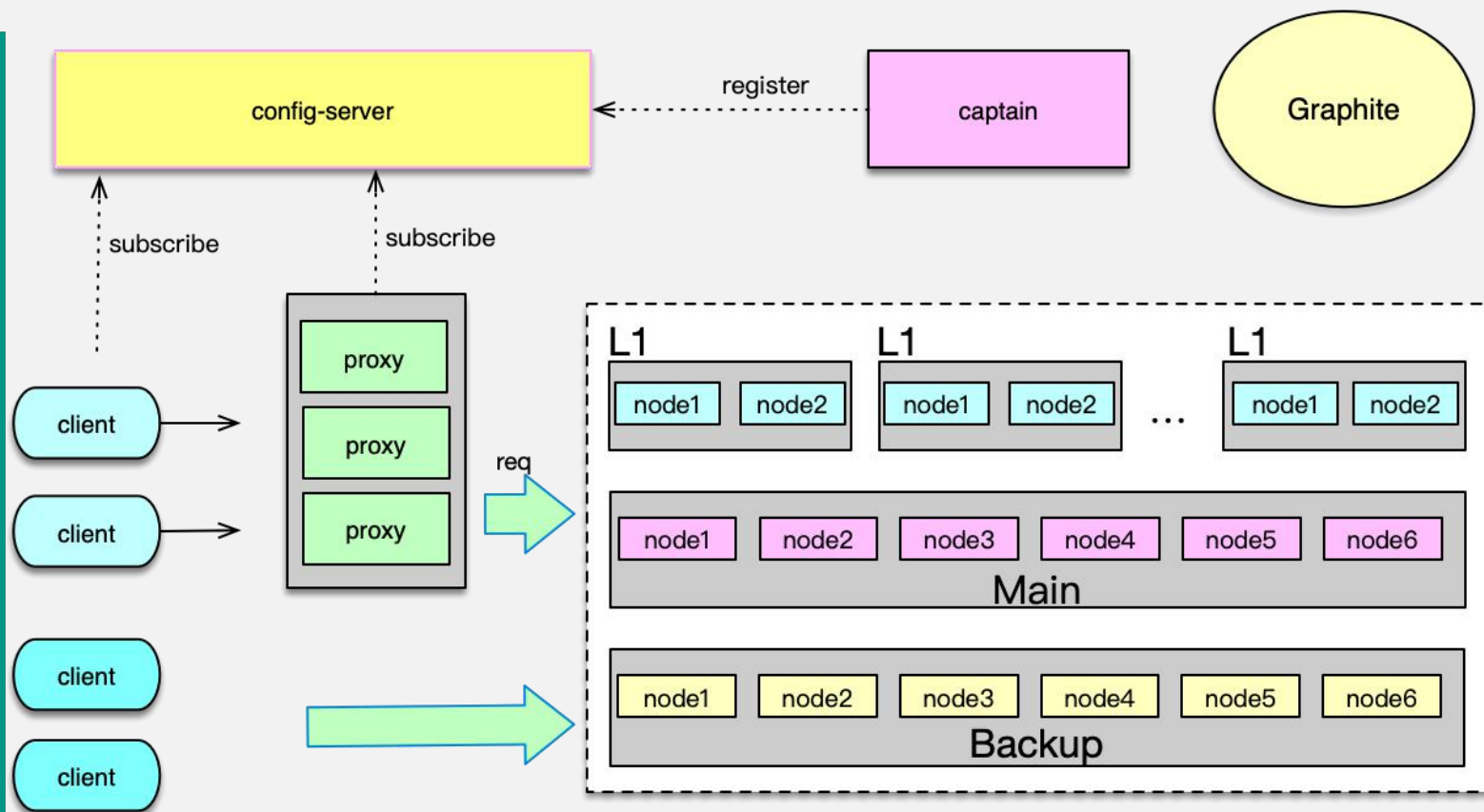
- Memcached 存储：L1-Main-Backup 架构
- L1 单池容量为Main 的1/10，L1一般有4-6组，存最热数据，抗流量洪峰
- Main 池容量最大，保存最近几乎所有较热的数据
- Backup池 容量一般为Main池的1/2以下，或者为其他IDC的Main 池，在Main池异常或miss后访问



社交Feed流缓存解析

Feed流 Mc 架构

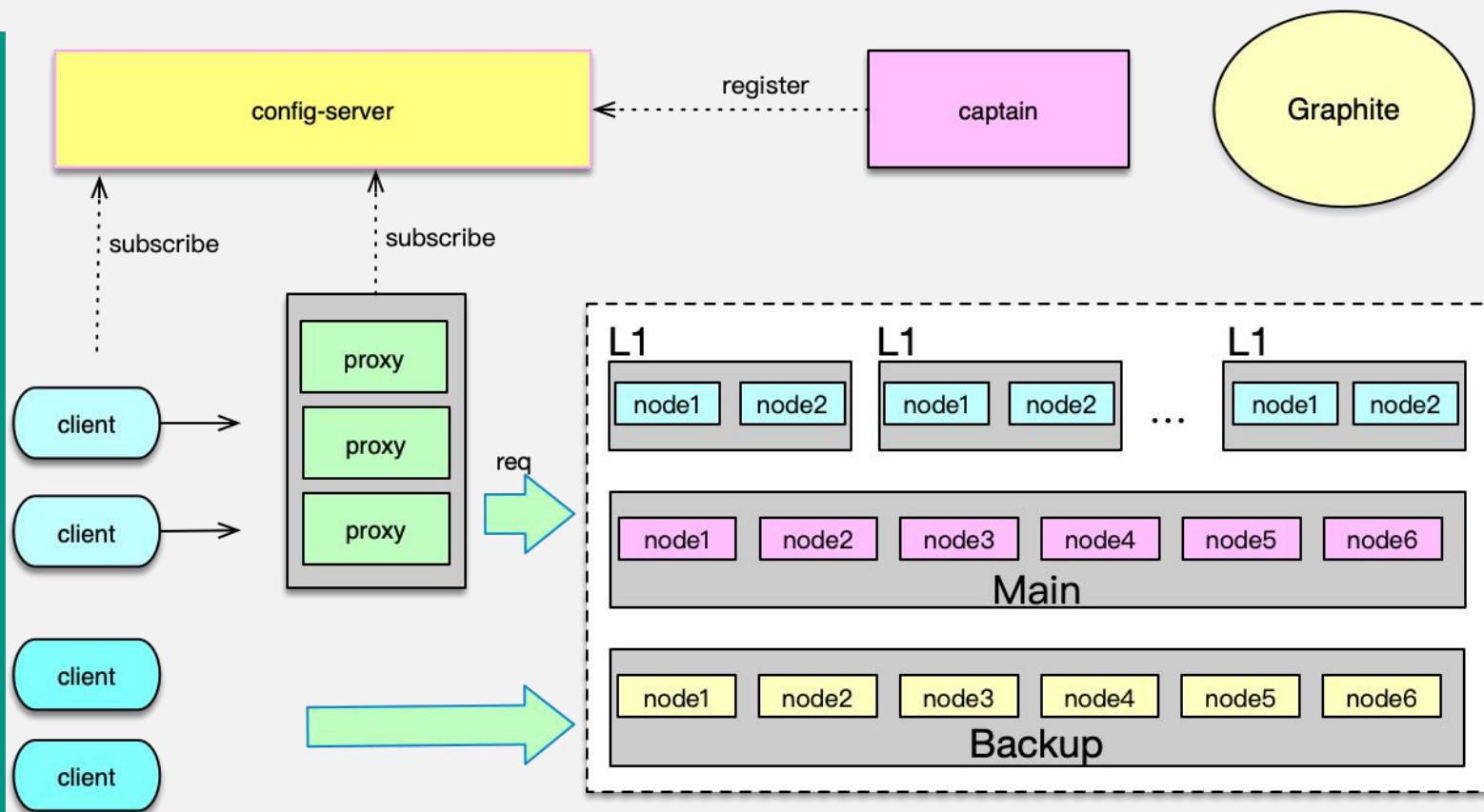
- proxy封装多层的访问逻辑，简化业务访问逻辑
- 业务也可以通过 Client可以直接访问mc多层缓存，相对复杂
- Captain 负责注册并修改mc部署及扩缩
- configServer 配置中心记录mc/proxy部署元数据，供proxy、client获取及感知变化
- 三层Mc架构，支持TPS 百万级，命中率 99%+，是Feed 流处理程序稳定运行的重要支撑



社交Feed流缓存解析

Feed流 Mc 架构

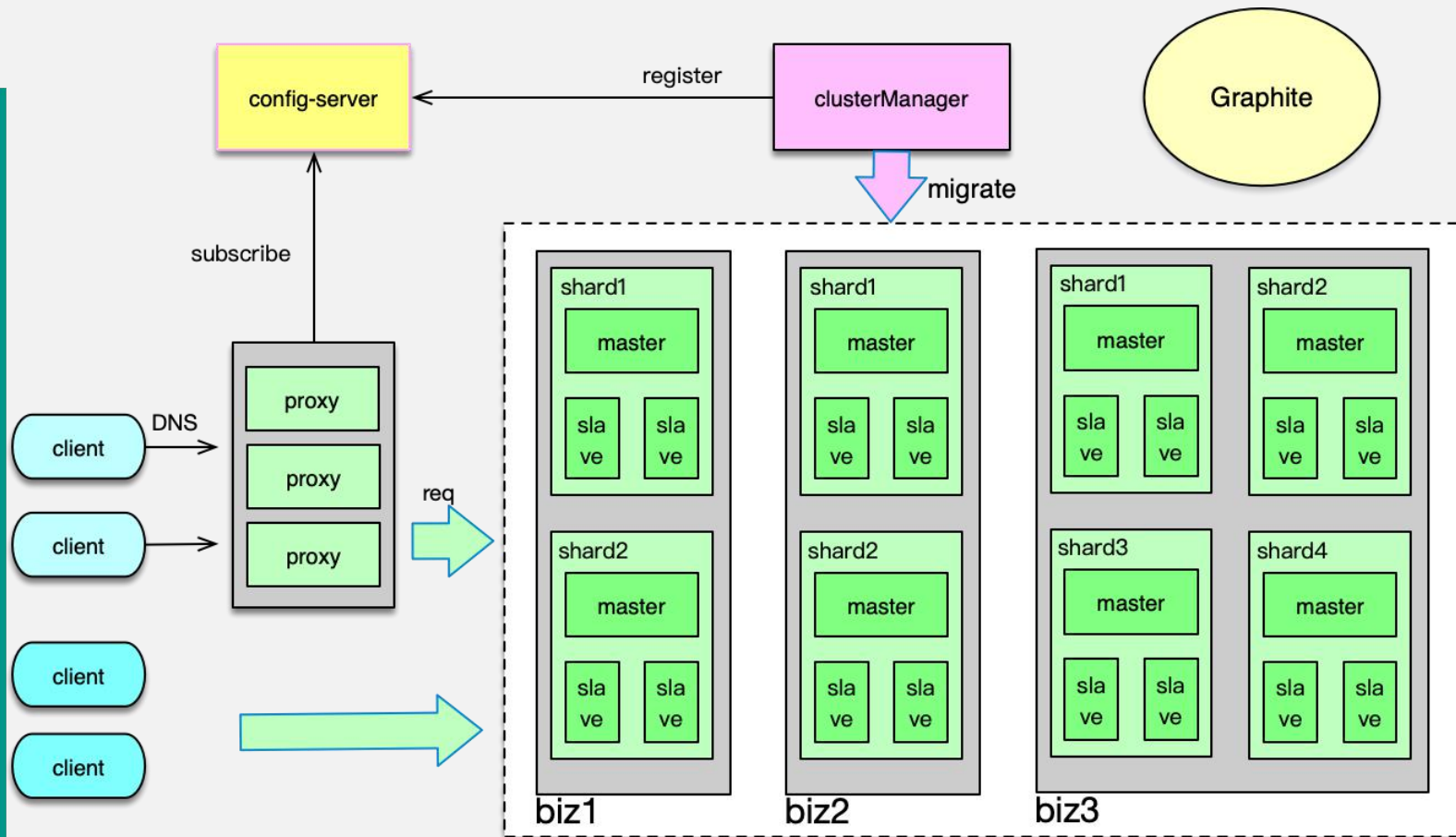
- 数据读取
 - 读取路径 L1-Main-Backup
 - 任何一层miss，下层命中后回种
 - 三层都miss，从Db加载后回种所有层
- 数据更新
 - 写路径 Main-Backup-L1以master-slave为基准
 - 支持 noreply，进行加速写入



社交Feed流缓存解析

Feed流 Redis 架构

- 单个子业务Redis部署1主多从
- 多个子业务按类型分为cluster集群，通过多租户proxy进行访问，不同子业务可以混合存储
- 性能考虑，支持smart client直接访问redis集群
- clusterManager 负责slot维护及迁移
- 配置中心记录业务集群相关的proxy、redis的配置及部署



感谢你对本专栏的支持
祝各位前程似锦！