

# 私塾在线 《高级软件架构师实战培训 阶段二》

# 跟着cc学架构系列精品教程

## 缓存规划-1

### n 提前说明

Web表现层的缓存在前面已经系统学习过了，这里主要关注逻辑层的缓存，简单的说，就是对逻辑层需要处理的数据的缓存

### n 缓存数据规划

缓存的数据有数据库数据、逻辑处理数据，有长期缓存、临时缓存等。常见需要考虑的基本点有：

- 1: 哪些数据要缓存
- 2: 数据的规模和特点，比如记录的条数，单条大小等
- 3: 数据缓存的时限，长期缓存和临时缓存
- 4: 数据被动失效和主动更新

### n 缓存的技术规划

- 1: 技术选型：JVM缓存、本地文件缓存、Memcached、Redis等
- 2: 性能规划：比如请求数、QPS（每秒查询率Query Per Second）等
- 3: 缓存的数据类型选择，这个要看具体的业务场景，要看如何使用这些数据

## 缓存规划-2

- 4: 缓存对象粒度，缓存的对象的数据库结构的设计
- 5: 是否需要集群，如何集群
- 6: 如何保证缓存数据的可用性
- 7: 如何保证缓存数据的一致性
- 8: 如何保证集群中数据的同步
- 9: 如何提高缓存命中率
- n 缓存部署环境规划
  - 1: 缓存的容量规划
  - 2: 缓存的部署环境，比如多少台机器，每台机器多少内存，网络带宽等
  - 3: 硬件环境、软件环境
  - 4: 复制、集群的部署方案
- n 可伸缩性，扩容的预定方案
- n 容灾规划

## Memcached集群-1

n 直接使用客户端实现

n magent介绍

magent是一款开源的Memcached代理服务器软件，其功能及特点：

- 1: 能和每个memcache保持多个长连接，可以减少memcache保持的连接数量，以及创建销毁连接的开销；不过，memcache本身就支持大并发连接，这个功能也就没什么特别的意义
- 2: 支持memcache的binary协议命令，实现请求的转发
- 3: 和memcache一样，基于libevent的事件驱动来处理IO
- 4: 支持ketama的一致性hash算法
- 5: 支持memcache backup集群，当memcache集群有机器down了，memagent会将get请求转向memcache backup集群

## Memcached集群-2

### n magent 安装

- 1: 使用的是magent0.6, 去<http://code.google.com/p/memagent/>下载
- 2: 需要先安装libevent, 前面课程已经安装过
- 3: 新建一个文件夹, 在里面解压tar包
- 4: /sbin/ldconfig
- 5: sed -i "s#LIBS = -levent#LIBS = -levent -lm#g" Makefile
- 6: make
- 7: 如果出现找不到文件的话, 编辑MakeFile文件, 修改LIBS的位置, 修改CFLAGS中include文件的位置
- 8: 如果还是找不到libevent的话, 去/usr/lib下查找, 如果有libevent存在, 那就是so文件的名称不对, 可以建一个软链: 如ln -s libevent-1.4.so.2 libevent.so
- 9: make成功的话, 会出现magent命令文件, 可以拷贝到/usr/bin下面去

## Memcached集群-3

### n Magent的基本用法

- 1: 先通过/usr/lib/magent -h 查看一下参数及其含义
- 2: 准备好主、备的Memcached
- 3: 启动magent, 如: /usr/bin/magent -u root -n 5120 -l 192.168.1.106 -p 11211 -s 192.168.1.106:2222 -s 192.168.1.106:1111 -b 192.168.1.106:3333
- 4: 然后就可以测试了

### n 使用Magent的注意事项

- 1: Magent相当于为Mem作了数据复制备份, 当Mem节点出现故障时, 能避免出现单点故障, 相当于实现了Mem的HA
- 2: Magent并不提供集群扩容的处理: 比如动态调整集群、比如数据自动迁移等
- 3: Magent每次写数据都会写到主和从上, 并且写的算法一样; 当主宕掉, magent会向从读取数据, 保证用户不受影响
- 4: 如果down掉的主Memcached重启的话, 此时该服务器没有数据, 因此magent取得的将是空值, 尽管备份Memcached还有数据, 也就是说Magent并不做数据的自动复制
- 5: Magent可以连接Magent, 所以可以使用它来模拟实现Mem的负载均衡