简单分布式缓存系统

【背景】

在云计算时代，全球的数据存储量以平均每年30%的速度不断增长。在这些数据中，用户访问的主要都是最近上传的热数据。作为存储系统应对热数据访问的有效补充，一个好的缓存系统就显得越来越重要。

【题目及考核点】

请设计和实现一个简单分布式缓存系统，其总体架构如下：

图形用户界面, 应用程序

描述已自动生成

系统总体上由一个Master，多个Cache Server，以及若干client组成。Master负责管理数据分布信息，每个Cache Server以LRU的形式缓存了一部分Key-Value数据。Client在读写数据之前，先查看本地数据分布缓存，如果有效，则根据key去访问对应的Cache Server；如果无效则需要重新从Master拉取分部信息。Cache Server定期向Master汇报存活信息，如果Cache Server故障，则心跳失效，Master会重新划分数据分布，并通知仍然存活的Cache Server这一决定。Cache Server的扩容和缩容逻辑与上面的描述类似。

**考察点：**

1. 单机LRU：单个Cache Server的LRU机制，只能保存至多指定数量的key，多余的按照LRU的机制淘汰
2. 分布式访问：Client会从Master拉取分布信息，并根据key来访问对应的Cache Server。
3. Cache Server扩容/缩容和容灾：支持Cache Server的容灾，以及Cache Server的扩容和缩容操作。
4. Master容灾：Master容灾较为复杂，可以不做，做了加分，要做的话，方案自己调研和设计。

【现场演示要求】

1. Client1随机产生等长Key和等长Value以稳定的压力持续写入缓存系统，写请求大致均匀访问各个Cache Server。
2. Client2加载指定的keylist文件，逐条向缓存系统读取key，能正确返回key在或者不在。
3. 单机Cache Server缓存超出限制后按照LRU机制淘汰Key，并保持内存使用稳定。
4. Cache Server故障剔除，Cache Server扩容和缩容

【交付要求】

1. 输出系统设计文档，介绍系统关键点（包括交互流程，接口及数据结构等）实现
2. 提交系统源代码（含单元测试），编程语言限定为C++(11标准或以上)
3. 在实验环境中部署运行，完成自测。

【环境部署&服务器要求说明】

1. 普通linux服务器或虚拟机，设备型号相同，分别部署以下服务角色：Client(1)、Master(1)，Cache Server1、Cache Server2、Cache Server3（如果资源受限，也可让所有服务角色运行在同一台服务器上，不影响系统逻辑和现场演示）