

实时排行-名次变化不大

比如曲谱排行，以点赞数排序，分数一次只能加1，名次变化小

离线刷排行

heap

toplist

步骤：

1 walk各表

对相关表进行walk，为每个榜单收集主要数据

对于数量不是很大的实体（比如帮派），同时全部加载到内存中

2 build

每个榜单计算最终的榜单（缺少info）

3 collect

从所有榜单中收集所有的角色、帮派等个体id

4 collect info

为上一步中收集到的各个实体完善信息

5 fill

为各个榜单填充info

6 save

将榜单保存到db

周期排行

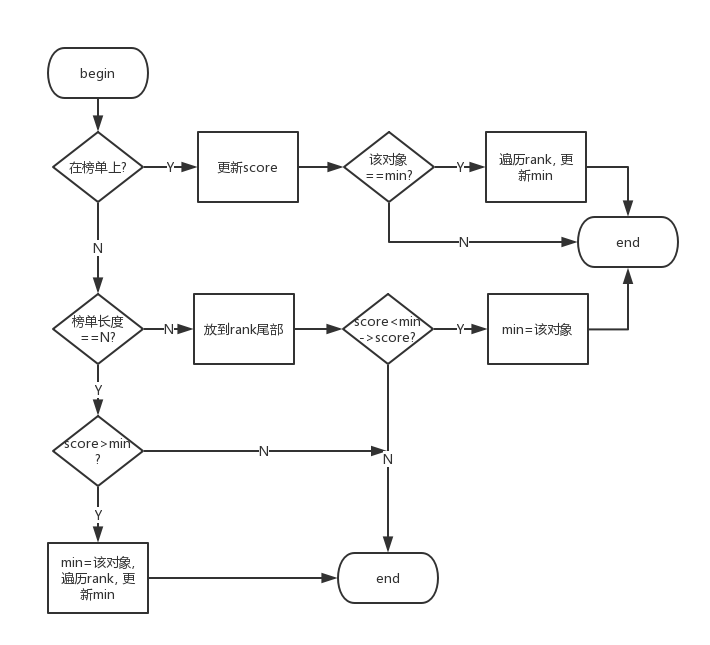
interval\_ranking

为了获得排名变化，服务器定时进行排序（比如10分钟一次），变化的少的话使用插入排序，变化的多用~~快排~~稳定排序

定时存储，停服重刷，关闭服务器要特别处理

程序崩溃如何处理？建议通过停服重刷，那等级榜可能在重刷后发生很大变化

score变小掉出排行榜导致尾部数据不准确



score可变小：

min==该对象 score变大 遍历rank，更新min

min==该对象 score没变大 更新score

min!=该对象 score变小 1比min->score小 min=该对象 2否则不处理

min!=该对象 score没变小 不处理

离线刷分布

ranking\_distribution

榜单之上的准确排名，榜单之外的近似排名

离线获取所有成员的排名分布，比如等级排行榜，分别统计每个等级有多少人，给出一个级别，就能获得大概有多少人等级更高。

问题：排名分布是否需要实时更新

冷启动问题

开始数据量少的时候，如何保证个人排名准确性

一个实体多种排行

程序启动时遍历一遍数据库，从中筛选出活跃对象，组成实时榜，该榜不存盘。

实时变化

方案一

struct RankInfo

{

int key;

int score;

int other\_info;

};

hashmap<int, RankInfo> rankMap;//<key,>

set<int> levelMap;//<level+key>

set<int> scoreMap;//<score+key>

从levelMap中获取排行榜还得再从rankMap获取详细数据

翻页不好弄

方案二

多个桶

score bucket1[]

score bucket2[]

score bucket3[]

score bucket4[]

hashmap<int, RankInfo> rankMap;

每个桶记有修改时间和排序时间

一个桶就是top n的方案二

一般来说玩家查看排行榜的话是从头开始看的

桶怎么划分？score范围？固定大小？

如果有两种排序方式，两种桶

方案三

一个桶

hashmap<int, RankInfo> rankMap;

int level[]

int score[]

其他

客户端负责去筛选

但是要客户端筛选就不能分页传输了

为这种可能会频繁变化的榜单设置一个计时或计数，或者弄一个缓存，同一玩家的数据累计到特定次数或特定时间才更新

gs也需要了解一些排行榜，比如有多少人等级到达多少级开飞升之类的

每次都向delivery查询是不合适的，能不能做一个订阅/通知系统，到达条件通知gs