redis实现时时直播列表缓存，支持分页[ 热点数据存储 ]

需求说明

在直播平台中，用户查看直播列表，这个列表数据具有时时性，并且是强热点数据，也就是说每时每刻都可能会有新的直播。这些数据如果每次都去数据库读取肯定是不可取的，所以这可以考虑使用redis来实现。

使用redis有序集合来保存直播列表

思路简介

由于每时每刻都有可能会有新的数据进入到sortedset中，所以使用redis的有序集合进行存储。之所以使用有序集合是考虑到翻页的时候不要有重复数据，如果需求允许重复数据的出现，则可以使用先进先出队列queue。

【重复数据出现原因：加入用户当前在看第一页的数据，此时又有两条新数据插入到sortedSet中，那么如果用” zrevrange(key, pageSize\*pageNum, pageSize\*(pageNum+1))”进行取值，就会出现2条重复数据了】

每生成一条数据，将其添加到redis有序集合中，根据ID( 如果id是数值型的 )或者时间戳（这个要进行处理，后面会讲）进行排序。取的时候通过以下方法进行取值。

Set<String> zrevrangeByScore(final String key, final double max, final double min, final int offset, final int count)

假如每页5条数据：

第一页：“jedisCluster.zrevrangeByScore(key, Double.MAX\_VALUE, 0, 0, 5)”

记录第一页中后一条数据的score是M

第二页：“jedisCluster.zrevrangeByScore(key, M, 0, 1, 5)”

以此类推。

【我们这里使用的jedisCluster客户端】

如何保证sortedSet集合数量

假如只缓存100条数据，每页10条显示。我们可以采取缓冲区策略，就像redis中保存字符串时候使用的SDS动态字符串。我们可以每次在保存的时候检测set的大侠zcard，如果zcard超过了120（允许多缓存20个，这样可以减少截取set的操作次数，提高效率），则可以使用以下命令进行截取至100：

jedisCluster.zremrangeByRank(key, 0, jedisCluster.zcard(key)-100)

并发安全

上面的策略存在并发安全问题，它包含两步：

1. 插入数据到set中。
2. 查询sortedSet大小，判断是否大于120
3. 如果大于120，则进行zremrangeByRank操作。

如果在高并发的环境中很容易多次执行zremrangeByRank，解决办法：

使用lua脚本，将上面操作写到lua脚本中。在lua脚本中执行的操作是原子性的，是线程安全的。不过要注意redis在执行lua脚本时是不可以处理其他请求的。

翻页超过了缓存大小

这种情况的话就可以考虑到数据库中查询，或者走搜索引擎，毕竟用户看到的最多的还是前几页的数据。具体要缓存多少数据要根据实际业务进行计算。