链滴

跨库跨表的分页查询实现

作者：changming
原文链接：https://ld246.com/article/1472302444980
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

对于数据库分库分表之后，涉及到查询时就会存在一些问题，比如如何分页，如何排序，如何处函数平均值等等，特别是对于分页功能，需要在应用中将数据合并后进行排序，在显示，还需要考虑用中翻页的页码与实际库中查询时页码的关系，同时还需要考虑某个库数据查询完毕后，其他库中如增加每次查询页码的问题，否则查询后的总页数不能满足应用中的页数。
当前GAP平台正在进行组织权限系统的服务化，存在一个统一集中式管理的远端组织权限服务系，同时也存在一个具体业务系统的本地组织权限，那么当用户登陆到本地业务系统后，在进行系统用管理时，希望看到的是能登陆到本地系统所有的用户信息，此时看到的用户数据来自远端（公共管理用户数据）和本地（业务系统特有的用户数据），如何实现分页？下面结合具体情况说明如何实现。
场景是2个数据库，要求排序，并且查询不是非常频繁，访问量也不会很大，没有用户会一直在里进行翻页操作，同时对于一个企业来说，用户信息频繁变动也是不切实际的。
对于分页功能来说，分两种情况，一种要求排序，一种不要求排序。非排序分页，可以按照同等长的方式在2个数据源上进行分页查询，也就是分页的每页记录数，平均来自这2个数据源，每次查询页的记录数都平均来自这2个数据源，当某个数据源没有符合条件的数据记录时，就需要单独查询另一个数据源，查询的记录数同时也就需要增加；而另外一种情况可以按照同等比例的分页方式，也就某个数据源每次分页查询时查询的记录数占该数据源符合条件的所有记录的比例是一定的，因为每次页查询查多少记录是根据2个数据源中所有符合条件的记录总数计算出来的，这样就保证了当分页查结束后，2个数据源也就同时查询完毕。
那么非排序分页的弊端在那里，首先不能进行排序，这样数据出来比较乱，用户不会满意，其次于同等比例或者同等步长这两种方式，需要考虑的因素比较多，如下：

每次分页如何平均2个数据源上的查询记录数
当每页记录数变化时怎么处理
当某个数据源查询完毕，没有符合条件的记录时，如何对另一个数据源增加每次查询的记录数？加后就会影响到计算起始记录数

所以我们决定选择排序分页方式。就是从2个数据源处将足够多的数据取回来，在应用中按照某条件进行归并排序，然后进行分页显示，难点在于取多少才算是足够的数据？考虑一种比较极端的情，就是满足当前页的排好序的记录都来自同一个数据源，那么就要求每次查询时，每个数据源的分页录数至少是这个值。比如当前每页显示4条记录，那么查询第一页时，需要每个数据源至少查出排好的记录数4条，也就是总共8条记录数，然后进行归并排序，再显示排好序的4条，另外4条记录将会被掉；当查询第二页时，是不是只要查询每个数据源中的第二页数据呢？其实不是这样的，因为虽然每数据源中的分页数据是已经排好序的，但是你无法确定这2个数据源中第二页数据之间的排序情况，可能其中一个数据源的第三页数据与另外一个数据源的第五页数据才是临近排序的，所以我们需要取每个数据源的前两页数据，再进行归并排序，那么当用户直接翻到100页的时候，就需要把每个数据的前100页所有数据都取出，然后进行排序，而且是每次分页查询都需要重新查询一次上次已经查出数据。
对于这种方式，我们的改进措施是使用缓存，将每次查询出的记录都保存在缓存中，当然是已经好序的。当用户第一次翻页时，查询第一页数据，实际是从2个数据源处总共查出4（这个偏移量是可配置的）页大小的数据（每个数据源2页大小数据），同时应用中排好序后进行缓存，这样这四页数全部从缓存中加载，速度很快；当翻到第五页的时候，再做一次查询（前四页数据不再重复查询），照偏移量大小，将取回来的数据与原来前四页数据进行重新排好序，然后再次放入缓存中，这样前8数据都是从缓存中获取，而且是排好序的。
极端情况是，用户一下子直接查询了第100页的数据，那么按照偏移量设置，每个数据源是查询200页的数据，总共是400页数据被一下子查询出来，然后进行排序放入缓存，对于这种情况，我们了预防措施，如果跨度过大，比如这里的100页翻页情况，我们直接不予进行查询。
通过结合缓存，保证了排序分页的实现，同时对每次查询条件不同，缓存的内容不同，所以系统于相同查询条件的查询结果都做了缓存，速度提升明显。并且当用户数据有更新时，并没有及时更新存，我们提供了一个刷新按钮，强制清除缓存，重新查询，因为缓存数据配置的是永不过期策略。</>