我说数据分页用Limit，面试官直接让我回去等消息

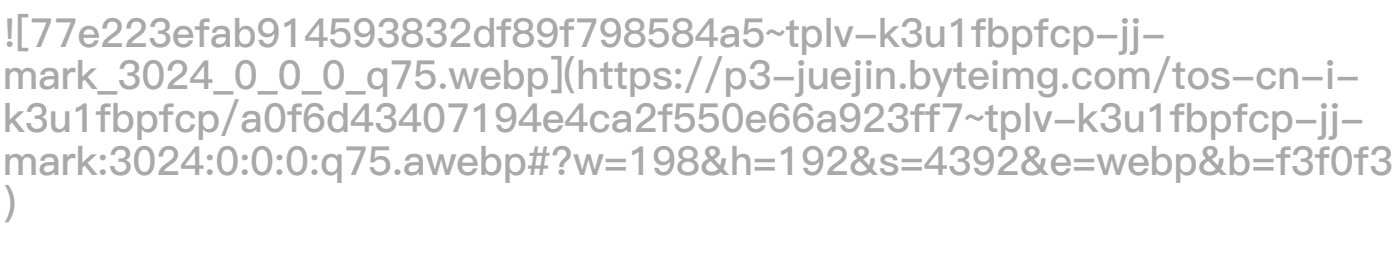
哈喽，大家哈，我是**DanMu**。最近我碰到了一个挺有趣的“小插曲”，大概是这样的：现在有一个社交应用，在聊天界面中，用户可以通过下滑页面来不断加载历史消息。我当时想不就一个分页，这么简单的需求怎么能难倒我这个练习时长两年半的SQL boy，我直接一个啪一个limit上去直接就把这个问题解决了，写出来的SQL大概是这样的：

```
select * from message order by create_time desc limit n_page*50 50;
```

看了两眼，好像没有什么问题，然后就跑去忙别的了。

问题浮现

等到了下午，我忙完了手上的其他需求，准备开始对这个功能进行一个简单的压测，然后就可以美美的把代码提交，去博德之门里扔几把骰子。但是随着压测的进行，我发现这个接口的响应时间波动比较剧烈，一开始我还没当回事，认为是数据库压力太大导致的性能下
降，但是紧接着我就发现了这些性能较差的接口都具有较大的`n_page`，同时普遍耗时都在其他接口的10倍以上



这一下可就不能当做没看见了（悲伤痛哭）。算了，发现问题那就解决问题，于是经过我的一番查找，很快我就发现，原来这是全是**深分页**惹的祸！

深分页问题

那么什么是深分页问题呢?
在`MySQL`的`limit`中：`limit 100,10`，`MySQL`会根据查询条件去存储引擎层找到前110条记录，然后在server层丢弃前100条记录取最后10条
这样先扫描完再丢弃的记录相当于白找，深分页问题指的就是这种场景（当limit的偏移量过大时会导致性能开销）。
那么要如何解决这个问题呢，现在比较多的实践是使用**游标分页。**

游标分页

游标分页的思想说起来也简单，既然采用`limit 100,10`这样的分页方式会导致MySQL从头开始扫描，从而扫描到没用的前100条记录，那么如果我们能够直接从第101条开始扫描，扫描10条，那么这个问题不就解决了。那么应该怎么实现呢，简单，在扫描时加一个条件不就得了

```
select * from message where create_time<上一次扫描到的最后一条记录的create_time order by create_time desc limit 10;
```

实际上游标翻页只是通过额外添加一个条件来改变MySQL扫描的起始位置。这也导致了游标翻页的使用必须要**满足两个前提**：

1. 需要一个列来记录上一次查询到的最大值，并且这个列要是有序的（比如我们上面用到的create_time）
2. 每一次的查询会依赖上一次查询的最大值，因此，分页查询需要时连续的，不能进行跳页（比如查完第一页然后跳过第二页直接查第三页）

接下来让我们来做一个简单实验，验证一下游标分页的有效性。为了凸显效果，我们直接给表里整进去100万数据（这会花费很多时间，你可以自行减少记录数量）

```
CREATE TABLE `test` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `empty_body` varchar(255) DEFAULT NULL,
  `create_time` datetime DEFAULT NOW(),
  PRIMARY KEY (`id`)
```

```
)ENGINE=MyISAM DEFAULT CHARSET=utf8 COMMENT='测试表';

DROP PROCEDURE IF EXISTS my_insert;

create procedure my_insert()
BEGIN
    declare i int default 1;
    set autocommit = 0;
    while i <= 1000000 do
            insert into diting.test
            (`empty_body`) values
                (concat('message',i));
            set i = i + 1;
        end while;
    set autocommit = 1;
END;

CALL my_insert();
```

接下来，让我们比较一下不同的查询方法所需要的时间

```
# 0.01 sec
select * from test limit 0,10;

# 深分页 0.28 sec
select * from test limit 1000000,10;

# 游标分页 0.00 sec
select * from test where id>1000000 limit 0,10;
```

可以看到，使用游标翻页的速度远远快于直接使用`limit`进行分页。

被忽略的问题

在上面我们证明了游标分页可以解决`limit`分页带来的性能问题，但是其实还有隐藏的，`limit`分页**无法解决**的问题：**消息重复问题**。让我们思考一下，如果我们在使用`limit`分页时，有新消息到来会怎么样。

k3u1fbpfcp/9a464e4c3b054ccc9f24f3d7cf5d05ed~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=3997&h=2135&s=473883&e=jpg&b=fdf0ea)
没错！由于`limit`分页是根据和最新记录的差值来计算记录位置的，所以如果有新消息到达，那么在读取下一页的时候，就有可能读取到和上一页重复的消息。但是由于游标翻页采用的是固定的游标，即使有新消息到达也不会读取到重复的消息。

总结
--

游标分页非常适合于深分页的场景，能够大大减少需要的记录数量，但是由于游标依赖于上一次查询，所以无法实现跳页，并且只有作为游标的列有序的情况下才能使用游标。而且这个解决方案可以很容易的迁移到项目中任何需要分页的地方，**非常适合作为面试时的技术亮点**。

点，不迷路

> 好了，以上就是这篇文章的全部内容了，如果你能看到这里，**非常感谢你的支持！**
> 如果你觉得这篇文章写的还不错， 求**点赞** 求**** 求**分享** 对暖男我来说真的 **非常有用！！！**
> 白嫖不好，创作不易，各位的支持和认可，就是我创作的最大动力，我们下篇文章见！
> 如果本篇博客有任何错误，请批评指教，不胜感激 ！

> 最后推荐我的**IM项目DiTing**（[github.com/danmuking/D...](http://cxyroad.com/”https://github.com/danmuking/DiTing-Go”)），致力于成为一个初学者友好、易于上手的 IM 解决方案，希望能给你的学习、面试带来一点帮助，如果人才你喜欢，给个Star叭！

原文链接: https://juejin.cn/post/7362046148708352035