

MySql limit 优化，百万至千万级快速分页

MySql 性能到底能有多高？用了 php 半年多，真正如此深入的去思考这个问题还是从前天开始。有过痛苦有过绝望，到现在充满信心！MySql 这个数据库绝对是适合 dba 级的高手去玩的，一般做一点1万篇新闻的小型系统怎么写都可以，用 xx 框架可以实现快速开发。可是数据量到了10万，百万至千万，他的性能还能那么高吗？一点小小的失误，可能造成整个系统的改写，甚至更本系统无法正常运行！好了，不那么多废话了。用事实说话，看例子：

数据表 collect (id, title ,info ,vtype) 就这4个字段，其中 title 用定长，info 用 text, id 是逐渐，vtype 是 tinyint，vtype 是索引。这是一个基本的新闻系统的简单模型。现在往里面填充数据，填充10万篇新闻。

最后 collect 为 10万条记录，数据库表占用硬盘1.6G。OK ,看下面这条 sql 语句：

select id,title from collect limit 1000,10; 很快；基本上0.01秒就 OK，再看下面的

select id,title from collect limit 90000,10; 从9万条开始分页，结果？

8-9秒完成，my god 哪出了问题？？？？其实要优化这条数据，网上找得到答案。看下面一条语句：

select id from collect order by id limit 90000,10; 很快，0.04秒就 OK。为什么？因为用了 id 主键做索引当然快。网上的改法是：

select id,title from collect where id>=(select id from collect order by id limit 90000,1) limit 10;

这就是用了 id 做索引的结果。可是问题复杂那么一点点，就完了。看下面的语句


select id from collect where vtype=1 order by id limit 90000,10; 很慢，用了8-9秒！

到了这里我相信很多人会和我一样，有崩溃感觉！vtype 做了索引了啊？怎么会慢呢？vtype 做了索引是不错，你直接 select id from collect where vtype=1 limit 1000,10; 是很快的，基本上0.05秒，可是提高90倍，从9万开始，那就是0.05*90=4.5秒的速度了。和测试结果8-9秒到了一个数量级。从这里开始有人提出了分表的思路，这个和 discuz 论坛是一样的思路。思路如下：

建一个索引表：t (id,title,vtype) 并设置成定长，然后做分页，分页出结果再到 collect 里面去找 info 。是否可行呢？实验下就知道了。

10万条记录到 t(id,title,vtype) 里，数据表大小20M 左右。用

select id from t where vtype=1 order by id limit 90000,10; 很快了。基本上0.1-0.2秒可以跑



完。为什么会这样呢？我猜想是因为 `collect` 数据太多，所以分页要跑很长的路。`limit` 完全和数据表的大小有关的。其实这样做还是全表扫描，只是因为数据量小，只有10万才快。OK，来个疯狂的实验，加到100万条，测试性能。

加了10倍的数据，马上 `t` 表就到了200多 M，而且是定长。还是刚才的查询语句，时间是0.1-0.2秒完成！分表性能没问题？错！因为我们的 `limit` 还是9万，所以快。给个大的，90万开始

`select id from t where vtype=1 order by id limit 900000,10;` 看看结果，时间是1-2秒！

why ?? 分表了时间还是这么长，非常之郁闷！有人说定长会提高 `limit` 的性能，开始我也以为，因为一条记录的长度是固定的，mysql 应该可以算出90万的位置才对啊？可是我们高估了 mysql 的智能，他不是商务数据库，事实证明定长和非定长对 `limit` 影响不大？怪不得有人说 `discuz` 到了100万条记录就会很慢，我相信这是真的，这个和数据库设计有关！

难道 MySQL 无法突破100万的限制吗？？到了100万的分页就真的到了极限？？

答案是：NO !!!! 为什么突破不了100万是因为不会设计 mysql 造成的。下面介绍非分表法，来个疯狂的测试！一张表搞定100万记录，并且10G 数据库，如何快速分页！

好了，我们的测试又回到 `collect` 表，开始测试结论是：30万数据，用分表法可行，超过30万他的速度会慢到你无法忍受！当然如果用分表+我这种方法，那是绝对完美的。但是用了我这种方法后，不用分表也可以完美解决！

答案就是：复合索引！有一次设计 mysql 索引的时候，无意中发现索引名字可以任取，可以选择几个字段进来，这有什么用呢？开始的 `select id from collect order by id limit 90000,10;` 这么快就是因为走了索引，可是如果加了 `where` 就不走索引了。抱着试试看的想法加了 `search(vtype,id)` 这样的索引。然后测试

`select id from collect where vtype=1 limit 90000,10;` 非常快！0.04秒完成！


再测试: `select id ,title from collect where vtype=1 limit 90000,10;` 非常遗憾，8-9秒，没走 `search` 索引！



再测试: `search(id,vtype)`，还是 `select id` 这个语句，也非常遗憾，0.5秒。

综上: 如果对于有 `where` 条件, 又想走索引用 `limit` 的, 必须设计一个索引, 将 `where` 放第一位, `limit` 用到的主键放第2位, 而且只能 `select` 主键!

完美解决了分页问题了。可以快速返回 `id` 就有希望优化 `limit`，按这样的逻辑，百万级的 `limit` 应该在0.0x 秒就可以分完。看来 mysql 语句的优化和索引时非常重要的！

好了，回到原题，如何将上面的研究成功快速应用于开发呢？如果用复合查询，我的轻量级框架就没的用了。分页字符串还得自己写，那多麻烦？这里再看一个例子，思路就出来了：





`select * from collect where id in (9000,12,50,7000);` 竟然 0秒就可以查完！

mygod , mysql 的索引竟然对于 in 语句同样有效！看来网上说 in 无法用索引是错误的！

有了这个结论，就可以很简单的应用于轻量级框架了：

代码如下：

```
$db=dblink();
$db->pagesize=20;

$sql="select id from collect where vtype=$vtype";


$db->execute($sql);
$strpage=$db->strpage(); //将分页字符串保存在临时变量，方便输出
while($rs=$db->fetch_array()){
    $strid.=$rs['id'].',';
}
$strid=substr($strid,0,strlen($strid)-1); //构造出 id 字符串
$db->pagesize=0; //很关键，在不注销类的情况下，将分页清空，这样只需要用一次数据库连接，不需要再开；
$db->execute("select id,title,url,sTime,gTime,vtype,tag from collect where id in ($strid)");



<?php while($rs=$db->fetch_array()): ?>
<tr>
    <td>&nbsp;<?php echo $rs['id'];?></td>
    <td>&nbsp;<?php echo $rs['url'];?></td>
    <td>&nbsp;<?php echo $rs['sTime'];?></td>
    <td>&nbsp;<?php echo $rs['gTime'];?></td>
    <td>&nbsp;<?php echo $rs['vtype'];?></td>
    <td>&nbsp;<a href="?act=show&id=<?php echo $rs['id'];?>" target="_blank"><?php
echo $rs['title'];?></a></td>
    <td>&nbsp;<?php echo $rs['tag'];?></td>
</tr>
<?php endwhile; ?>
</table>
<?php
echo $strpage;
```

通过简单的变换，其实思路很简单：1)通过优化索引，找出 id，并拼成 "123,90000,12000" 这样的字符串。2) 第2次查询找出结果。

小小的索引+一点点的改动就使 mysql 可以支持百万甚至千万级的高效分页！

通过这里的例子，我反思了一点：对于大型系统，PHP 千万不能用框架，尤其是那种连





sql 语句都看不到的框架！因为开始对于我的轻量级框架都差点崩溃！只适合小型应用的快速开发，对于 ERP,OA，大型网站，数据层包括逻辑层的东西都不能用框架。如果程序员失去了对 sql 语句的把控，那项目的风险将会成几何级数增加！尤其是用 mysql 的时候，mysql 一定需要专业的 dba 才可以发挥他的最佳性能。一个索引所造成的性能差别可能是上千倍！

PS: 经过实际测试，到了100万的数据，160万数据，15G 表，190M 索引，就算走索引，limit 都得0.49秒。所以**分页最好别让别人看到10万条以后的数据**，要不然会很慢！就算用索引。经过这样的优化，mysql 到了百万级分页是个极限！但有这样的成绩已经很不错，如果你是用 sqlserver 肯定卡死！而160万的数据用 id in (str) 很快，基本还是0秒。如果这样，千万级的数据，mysql 应该也很容易应付。

