

willko

浏览: 96371 次

性别: 

来自: 深圳

 我现在离线

 [详细资料](#)  [留言簿](#)

搜索本博客

最近访客 [>>更多访客](#)



[DesHpoT](#)



[niubo_](#)



[haoningabc](#)



[woodmoon](#)

博客分类

- [全部博客 \(55\)](#)
- [Javascript & Web \(10\)](#)
- [Linux \(4\)](#)
- [MySQL & Sphinx \(14\)](#)
- [nginx \(6\)](#)
- [php \(21\)](#)
- [Squid & NCache & Varnish \(0\)](#)
- [Cache & Storage \(11\)](#)
- [LVS & HAProxy \(0\)](#)
- [MySQL Proxy & Amoeba \(0\)](#)
- [python \(0\)](#)
- [erlang \(0\)](#)
- [c & cpp \(0\)](#)
- [ifs & moosefs \(0\)](#)

2009-02-09

◀ [Tokyo Tyrant 在 php 上不能自动反序列化...](#) | [安装mysql5 innodb存储引擎](#) ▶

 6顶
 2踩

[mysql分页limit 优化](#)

博客分类: [MySQL & Sphinx](#)

[MySQL](#) [SQL](#) [算法](#) [Blog](#) [.net](#)


mysql的分页比较简单，只需要limit offset,length就可以获取数据了，但是当offset和length比较大的时候，mysql明显性能下降

1.子查询优化法

先找出第一条数据，然后大于等于这条数据的id就是要获取的数据

缺点：数据必须是连续的，可以说不能有where条件，where条件会筛选数据，导致数据失去连续性

实验下

Sql代码 

```
1. mysql> set profiling=1;
2. Query OK, 0 rows affected (0.00 sec)
3.
4. mysql> select count(*) from Member;
5. +-----+
6. | count(*) |
7. +-----+
8. | 169566 |
9. +-----+
10. 1 row in set (0.00 sec)
11.
12. mysql> pager grep !~-
13. PAGER set to 'grep !~-'
14.
15. mysql> select * from Member limit 10, 100;
16. 100 rows in set (0.00 sec)
17.
18. mysql> select * from Member where MemberID >= (select MemberID from Member limit 10,1) limit 10
19. 100 rows in set (0.00 sec)
20.
21. mysql> select * from Member limit 1000, 100;
22. 100 rows in set (0.01 sec)
23.
24. mysql> select * from Member where MemberID >= (select MemberID from Member limit 1000,1) limit
25. 100 rows in set (0.00 sec)
26.
27. mysql> select * from Member limit 100000, 100;
28. 100 rows in set (0.10 sec)
29.
30. mysql> select * from Member where MemberID >= (select MemberID from Member limit 100000,1) limi
31. 100 rows in set (0.02 sec)
32.
33. mysql> nopager
34. PAGER set to stdout
```

我的留言簿 [>>更多留言](#)

■ 竟然很多都是半拉文章，没法看下去.....

-- by [thesp2](#)

■ 请教如何在客户端计算上传进度？

-- by [yangjie191191](#)

■ dd

-- by [xiaoyunlove](#)

其他分类

■ [我的收藏](#) (2)

■ [我的代码](#) (0)

■ [我的论坛主题帖](#) (8)

■ [我的所有论坛帖](#) (185)

■ [我的精华良好帖](#) (0)

■ [我解决的问题](#) (1)

最近加入群组


存档


■ [2010-07](#) (1)

■ [2010-06](#) (1)

■ [2010-05](#) (3)

■ [更多存档...](#)

 [RSS](#)



```
35.
36.
37. mysql> show profiles\G
38. ***** 1. row *****
39. Query_ID: 1
40. Duration: 0.00003300
41. Query: select count(*) from Member
42.
43. ***** 2. row *****
44. Query_ID: 2
45. Duration: 0.00167000
46. Query: select * from Member limit 10, 100
47. ***** 3. row *****
48. Query_ID: 3
49. Duration: 0.00112400
50. Query: select * from Member where MemberID >= (select MemberID from Member limit 10,1) limit
51. 100
52. ***** 4. row *****
53. Query_ID: 4
54. Duration: 0.00263200
55. Query: select * from Member limit 1000, 100
56. ***** 5. row *****
57. Query_ID: 5
58. Duration: 0.00134000
59. Query: select * from Member where MemberID >= (select MemberID from Member limit 1000,1) lim
60. it 100
61. ***** 6. row *****
62. Query_ID: 6
63. Duration: 0.09956700
64. Query: select * from Member limit 100000, 100
65. ***** 7. row *****
66. Query_ID: 7
67. Duration: 0.02447700
68. Query: select * from Member where MemberID >= (select MemberID from Member limit 100000,1) l
69. imit 100
```

从结果中可以得知，当偏移1000以上使用子查询法可以有效的提高性能。

2.倒排表优化法

倒排表法类似建立索引，用一张表来维护页数，然后通过高效的连接得到数据

缺点：只适合数据数固定的情况，数据不能删除，维护页表困难

具体请看，http://blog.chinaunix.net/u/29134/showart_1333566.html

3.反向查找优化法

当偏移超过一半记录数的时候，先用排序，这样偏移就反转了

缺点：order by优化比较麻烦，要增加索引，索引影响数据的修改效率，并且要知道总记录数，偏移大于数据的一半


引用

limit偏移算法：
正向查找： (当前页 - 1) * 页长度
反向查找： 总记录 - 当前页 * 页长度

做下实验，看看性能如何

总记录数： 1,628,775
每页记录数： 40
总页数： 1,628,775 / 40 = 40720
中间页数： 40720 / 2 = 20360


第21000页
正向查找SQL:

Sql代码 

```
1. | SELECT * FROM `abc` WHERE `BatchID` = 123 LIMIT 839960, 40
```

时间： 1.8696 秒


反向查找sql:

Sql代码 

```
1. | SELECT * FROM `abc` WHERE `BatchID` = 123 ORDER BY InputDate DESC LIMIT 788775, 40
```

时间： 1.8336 秒


第30000页
正向查找SQL:

Sql代码 

```
1. | SELECT * FROM `abc` WHERE `BatchID` = 123 LIMIT 1199960, 40
```

时间： 2.6493 秒

反向查找sql:

Sql代码 

```
1. | SELECT * FROM `abc` WHERE `BatchID` = 123 ORDER BY InputDate DESC LIMIT 428775, 40
```

时间： 1.0035 秒

注意，反向查找的结果是是降序desc的，并且InputDate是记录的插入时间，也可以用主键联合索引，但是不方便。

4.limit限制优化法
把limit偏移量限制低于某个数。。超过这个数等于没数据，我记得alibaba的dba说过他们是这样做的

5.只查索引法

<http://willko.iteye.com/blog/670120>

总结：**limit**的优化限制都比较多，所以实际情况用或者不用只能具体情况具体分析了。页数那么后，基本很少人看的。。。

6

顶

2

踩

分享到：





◀ [Tokyo Tyrant 在 php 上不能自动反序列化 ...](#) | [安装mysql5 innodb存储引擎](#) ▶

12:55 | [评论 / 浏览 \(1 / 9798\)](#) | 分类:[数据库](#) | [相关推荐](#) [▶ MORE](#)

评论

1 楼 [cevin](#) 2010-08-08

引用

总结：**limit**的优化限制都比较多，所以实际情况用或者不用只能具体情况具体分析了。页数那么后，基本很少人看的。。。

对于论坛来说,前3页和最后5页是浏览量最高的.如果对实时性要求不高,可以考虑做缓存.(Dz的缓存做的很人性化).反之,则是优化的重点.😁😁

发表评论



[您还没有登录,请您登录后再发表评论](#)