

穆晟铭

<

2018年5月

>

日

一

二

三

四

五

六

29

30

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

1

2

3

4

5

6

7

8

9

昵称: 穆晟铭

园龄: 4年11个月

粉丝: 6

关注: 10

+加关注

搜索

找找看

谷歌搜索

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

更多链接

我的标签

yii(4)

随笔分类

编辑器(1)

centos(39)

git(4)

java(93)

js(jquery)(18)

jsp(23)

mysql(16)

oracle(14)

php(143)

spring(43)

spring mvc(35)

sprint boot(16)

其他(10)

生活随笔(4)

数据库

笑话

随笔档案

2018年5月 (27)

2018年4月 (17)

2018年3月 (12)

2018年2月 (6)

## mysql的SQL\_CALC\_FOUND\_ROWS 使用 类似count(\*) 使用性能更高

mysql的SQL\_CALC\_FOUND\_ROWS 使用 类似count(\*) 使用性能更高

```
1 在很多分页的程序中都这样写：
2  SELECT COUNT(*) from `table` WHERE .....;  查出符合条件的记录总数
3  SELECT * FROM `table` WHERE ..... limit M,N;  查询当页要显示的数据
4  这样的语句可以改成：
5  SELECT SQL_CALC_FOUND_ROWS * FROM `table` WHERE ..... limit M, N;
6  SELECT FOUND_ROWS();
7  这样只要执行一次较耗时的复杂查询可以同时得到与不带limit同样的记录条数
8  第二个 SELECT返回一个数字，指示了在没有LIMIT子句的情况下，第一个SELECT返回了多少行（若上述的 SELECT语句不包括 SQL
```

http://blog.csdn.net/cuew1987/article/details/17393319

1.创建表:

```
1  //此处使用了覆盖索引
2  CREATE TABLE IF NOT EXISTS `Ben` (
3  `aa` int(10) unsigned NOT NULL AUTO_INCREMENT,
4  `bb` int(10) unsigned NOT NULL,
5  `cc` varchar(100) NOT NULL,
6  PRIMARY KEY (`aa`),
7  KEY `bar` (`bb`,`aa`)
8  ) ENGINE=MyISAM;
9
10 //无覆盖索引
11 DROP TABLE IF EXISTS `ben`;
12 CREATE TABLE IF NOT EXISTS `ben` (
13 `aa` int(10) unsigned NOT NULL AUTO_INCREMENT,
14 `bb` int(10) unsigned NOT NULL,
15 `cc` varchar(100) NOT NULL,
16 PRIMARY KEY (`aa`),
17 KEY `bar` (`bb`)
18 ) ENGINE=MyISAM;
```

2.插入100万条数据:

```
1  //插入数据
2  function insertData(){
3      //数据表中插入100万条数据
4      $ben = new Model();
5      for($i = 0; $i < 1000000 ; $i++){
6          $a = $i % 2 == 0 ? 1 : 0;
```

2018年1月 (47)  
2017年12月 (25)  
2017年11月 (9)  
2017年10月 (9)  
2017年9月 (20)  
2017年8月 (9)  
2017年7月 (17)  
2017年6月 (9)  
2017年5月 (17)  
2017年4月 (22)  
2017年3月 (18)  
2017年2月 (6)  
2017年1月 (7)  
2016年12月 (8)  
2016年11月 (10)  
2016年10月 (24)  
2016年9月 (20)  
2016年8月 (6)  
2016年7月 (2)  
2015年11月 (2)  
2015年10月 (3)  
2015年9月 (1)  
2015年8月 (3)  
2015年7月 (3)  
2015年5月 (1)  
2015年4月 (5)  
2015年3月 (4)  
2015年1月 (2)  
2014年11月 (1)  
2014年9月 (6)  
2014年8月 (7)  
2014年7月 (3)  
2014年6月 (3)  
2014年5月 (1)  
2014年3月 (1)  
2014年2月 (1)  
2013年12月 (1)  
2013年11月 (3)  
2013年10月 (4)  
2013年7月 (14)  
2013年6月 (19)

运维博客推荐

最新评论

1. Re:java错误: The supercl  
ass "javax.servlet.http.Http  
Servlet" was not found on t  
he Java Bu  
6  
--Captain\_Li
2. Re:phalcon: acl权限控制  
非常给力~ 学习ing  
--unaj
3. Re:java错误: The supercl  
ass "javax.servlet.http.Http  
Servlet" was not found on t  
he Java Bu  
@MISS\_CM、项目里面也需要  
引入。...  
--穆晟铭
4. Re:java错误: The supercl  
ass "javax.servlet.http.Http  
Servlet" was not found on t  
he Java Bu

```
7         $sql="insert into ben values (null, ".$a.", '".md5($i)."' );  
8         $aa = $ben->query($sql);  
9     }  
10 }
```

3.读取数据，测试程序执行时间：

```
1     function read01(){  
2         $start = microtime(true);  
3         $ben = new Model();  
4         $ben->query("select count(*) from ben where bb=1");  
5         $ben->query("select aa from ben where bb=1 limit 100,10");  
6         $end = microtime(true);  
7         echo $end - $start; //覆盖索引: 0.018204927444458, 无覆盖索引: 0.017701148986816  
8     }  
9     function read02(){  
10        $start = microtime(true);  
11        $ben = new Model();  
12        $ben->query("select SQL_CALC_FOUND_ROWS aa from ben where bb=1 limit 100,10");  
13        $ben->query("select FOUND_ROWS()");  
14        $end = microtime(true);  
15        echo $end - $start; //覆盖索引: 0.017460823059082, 无覆盖索引: 0.20762395858765  
16    }
```

4.结论：

使用覆盖索引情况下，使用SQL\_CALC\_FOUND\_ROWS性能较高；无覆盖索引情况下，使用count(\*)性能较高。另使用innoDB引擎比MyISAM低。

附：

资料一：覆盖索引

select aa from ben where bb = 1;

1)统计总数：

select count(\*) from ben where bb=1;

当我们在bb建立索引后，这个查询使用的就是覆盖索引（Covering Index）。

2)在分页操作中：

select id,title,content from ben order by createtTime desc limit 10000,10;

通常这样的查询会把索引建在createdTime字段（其中id是主键），不过当limit偏移很大时，查询效率仍然很低,采用覆盖索引来提升性能：

CREATE INDEX indexName ON ben ('createTime','id')，可以使用explain 查看是否是覆盖索引，出现extra:using index 就是

资料二：引擎MyISAM 和 InnoDB区别

这样做了，还是不可以  
--MISS\_CM、  
5. Re: 一个表单里，如果有<button>标签存在，它会自动提交表单  
我也是刚发现，差点怀疑自己了，，，简直坑，，，，  
--老男孩733

阅读排行榜

- 1. java错误: The superclass "javax.servlet.http.HttpServlet" was not found on the Java Bu(19820)
- 2. php正则: 匹配(), {}, [] 小括号, 大括号, 中括号里面的内容(6998)
- 3. 微信公共平台php用\$GLOBALS["HTTP\_RAW\_POST\_DATA"]收不到信息解决方法(5686)
- 4. yii2: 多条件多where条件下碰到between时, between语句如何处理呢? (4119)
- 5. phalcon: 获取参数的方法(4094)

评论排行榜

- 1. java错误: The superclass "javax.servlet.http.HttpServlet" was not found on the Java Bu(3)
- 2. 一个表单里，如果有<button>标签存在，它会自动提交表单(1)
- 3. phalcon: acl权限控制(1)
- 4. wampserver2.5 apache2.4.9: forbidden,本机可以访问,局域网内部能访问。(1)
- 5. phalcon:数据库分库,读写分离,负载均衡 系统方法执行顺序(1)

推荐排行榜

- 1. java错误: The superclass "javax.servlet.http.HttpServlet" was not found on the Java Bu(2)
- 2. java: jsp:param中文乱码(1)
- 3. mysql:联合索引及优化(1)
- 4. 'yii\base\InvalidRouteException' with message 'Unable to resolve the request "site/error"'.(1)
- 5. git: windows git ssh keys生成(1)

InnoDB和MyISAM是许多人在使用MySQL时最常用的两个表类型，这两个表类型各有优劣，视具体应用而定。基本的差别为：  
MyISAM类型不支持事务处理等高级处理，而InnoDB类型支持。

MyISAM类型的表强调的是性能，其执行速度比InnoDB类型更快，但是不提供事务支持，而InnoDB提供事务支持已经外部键等高级数据库功能。

以下是一些细节和具体实现的差别：

- ◆1.InnoDB不支持FULLTEXT类型的索引。
- ◆2.InnoDB 中不保存表的具体行数，也就是说，执行select count(\*) from table时，InnoDB要扫描一遍整个表来计算有多少行，但是MyISAM只要简单的读出保存好的行数即可。注意的是
- ◆3.对于AUTO\_INCREMENT类型的字段，InnoDB中必须包含只有该字段的索引，但是在MyISAM表中，可以和其他字段一起建立联合索引。
- ◆4.DELETE FROM table时，InnoDB不会重新建立表，而是一行一行的删除。
- ◆5.LOAD TABLE FROM MASTER操作对InnoDB是不起作用的，解决方法是首先把InnoDB表改成MyISAM表，导入数据后再改成InnoDB表，但是对于使用的额外的InnoDB特性(例如外键)的表不

适用。

另外，InnoDB表的行锁也不是绝对的，假如在执行一个SQL语句时MySQL不能确定要扫描的范围，InnoDB表同样会锁全表，例如update table set num=1 where name like “%aaa%”

两种类型最主要的差别就是Innodb 支持事务处理与外键和行级锁.而MyISAM不支持.所以MyISAM往往就容易被人认为只适合在小项目中使用。

我作为使用MySQL的用户角度出发，Innodb和MyISAM都是比较喜欢的，但是从我目前运维的数据库平台要达到需求：99.9%的稳定性，方便的扩展性和高可用性来说的话，MyISAM绝对是我

的首选。

原因如下：

- 1、首先我目前平台上承载的大部分项目是读多写少的项目，而MyISAM的读性能是比Innodb强不少的。
- 2、MyISAM的索引和数据是分开的，并且索引是有压缩的，内存使用率就对应提高了不少。能加载更多索引，而Innodb是索引和数据是紧密捆绑的，没有使用压缩从而会造成Innodb比

MyISAM体积庞大不小。

3、从平台角度来说，经常隔1，2个月就会发生应用开发人员不小心update一个表where写的范围不对，导致这个表没法正常用了，这个时候MyISAM的优越性就体现出来了，随便从当天拷

贝的压缩包取出对应表的文件，随便放到一个数据库目录下，然后dump成sql再导回到主库，并把对应的binlog补上。如果是Innodb，恐怕不可能有这么快速度，别和我说让Innodb定期用导出

xxx.sql机制备份，因为我平台上最小的一个数据库实例的数据量基本都是几十G大小。

4、从我接触的应用逻辑来说，select count(\*) 和order by 是最频繁的，大概能占了整个sql总语句的60%以上的操作，而这种操作Innodb其实也是会锁表的，很多人以为Innodb是行级

锁，那个只是where对它主键是有效，非主键的都会锁全表的。

5、还有就是经常有很多应用部门需要我给他们定期某些表的数据，MyISAM的话很方便，只要发给他们对应那表的frm.MYD,MYI的文件，让他们自己在对应版本的数据库启动就行，而

Innodb就需要导出xxx.sql了，因为光给别人文件，受字典数据文件的影响，对方是无法使用的。

6、如果和MyISAM比insert写操作的话，Innodb还达不到MyISAM的写性能，如果是针对基于索引的update操作，虽然MyISAM可能会逊色Innodb,但是那么高并发的写，从库能否追的上也是

一个问题，还不如通过多实例分库分表架构来解决。

7、如果是用MyISAM的话，merge引擎可以大大加快应用部门的开发速度，他们只要对这个merge表做一些select count(\*)操作，非常适合大项目总量约几亿的rows某一类型(如日志，调查

统计)的业务表。

当然Innodb也不是绝对不用，用事务的项目如模拟炒股项目，我就是用Innodb的，活跃用户20多万时候，也是很轻松应付了，因此我个人也是很喜欢Innodb的，只是如果从数据库平台应

用出发，我还是会首选MyISAM。

另外，可能有人会说你MyISAM无法抗太多写操作，但是我可以通过架构来弥补，说个我现有用的数据库平台容量：主从数据总量在几百T以上，每天十多亿 pv的动态页面，还有几个大项

目是通过数据接口方式调用未算进pv总数，(其中包括一个大项目因为初期memcached没部署,导致单台数据库每天处理 9千万的查询)。而我的整体数据库服务器平均负载都在0.5-1左右。

转载请注明出处，谢谢！

分类: [mysql](#)

好文要顶

关注我

收藏该文

穆晟铭

关注 - 10

粉丝 - 6

+加关注

0

0

« 上一篇: [java: jsp:param中文乱码](#)  
» 下一篇: [php/js/linux: js加密（rsa公钥加密） php解密（rsa私钥解密）](#)

posted @ 2017-11-07 17:07 穆晟铭 阅读(626) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#)网站首页。





## 如何快速搭建微信小程序？

一站式配置主机和域名 套餐11元/月起

立即抢购



最新IT新闻：

- 余晟：软件工程，“我们不一样”？
  - 微视努力撕掉抄袭标签 能否复制抖音增长轨迹？
  - 小米估值争议背后，缺失技术的IoT如何撑起互联网之梦
  - 放弃深度学习？我承认是因为线性代数
  - 链家人事大动荡 为上市不顾一切
- » 更多新闻...



云数据库 MySQL 5.7

高性价比单机版

84元/月



最新知识库文章：

- 你可以把编程当做一项托付终身的职业
  - 评审的艺术——谈谈现实中的代码评审
  - 如何高效学习
  - 如何成为优秀的程序员？
  - 菜鸟工程师的超神之路 -- 从校园到职场
- » 更多知识库文章...

历史上的今天：

2016-11-07 [MySQL](#)按照汉字的拼音排序,[mysql](#)汉字排序