链滴

当 mysql 表从压缩表变成普通表会发生什么？

作者：whimsy

原文链接：https://ld246.com/article/1657989584384

来源网站：链滴

前言

本文章做了把mysql表从压缩表过渡到普通表的实验过程，看看压缩表变成普通表会发生什么？本文对mysql5.7和mysql8分别进行了实验。

1、什么是表压缩

在介绍压缩表变成普通表前，首先给大家普及下，什么是表压缩。

表压缩，意思是使表中的数据以压缩格式存储，压缩能够显著提高处理速度和压缩磁盘 。压缩意味着硬盘和内存之间传输的数据更小且占用相对少的内存及硬盘，对于辅助索引，这种压缩带来更加明显好处，因为索引数据也被压缩了。

** 表压缩是有很大好处的，能减少磁盘的I/O，还能提高系统吞吐量，节约空间，压缩率越大，占用磁盘空间越小，文件传输时间提升，降低数据的存储和网络传输成本。 **

2、如何表压缩(mysql的版本需要大于5.5)

1、首先设置my.inf参数

```
#打开配置文件
vim /etc/my.inf

#加入配置项
innodb_file_per_table=1
innodb_file_format=Barracuda
innodb_strict_mode=1 #建议加上
innodb_default_row_format = COMPRESSED #在整个库默认启用行压缩格式时设定，一边不改变值

#重启数据库
systemctl restart mysqld
```

2、对表压缩

```
mysql> alter table t1 ROW_FORMAT=COMPRESSED;
```

3、压缩表转换为普通表

```
mysql> alter table t1 ROW_FORMAT=DEFAULT;
```

针对mysql5.7开始实验

- **mysql数据库版本：5.7.31**
- **linux版本：centos5.7**

1、建表和初始化测试数据

```
#1、建表
CREATE TABLE test_compress (
    id bigint(20) unsigned NOT NULL,
    identification_id int(10) unsigned DEFAULT NULL,
    timestamp datetime NOT NULL,
    action varchar(50) NOT NULL,
    result varchar(50) NOT NULL,
    PRIMARY KEY (id),
    KEY INDEX_test_compress_result (result),
    KEY INDEX_test_compress_timestamp (timestamp)
    ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

#2、插入测试数据（linux里执行脚本）
for NUM in {1..100000}; do mysql -h localhost PS_57 -e "insert into test_compress (id, identific
tion_id, timestamp, action, result) values ($NUM,$NUM*100,now(),concat('string',$NUM),conc
t('VeryVeryLargeString',$NUM))"; done
```

2、验证表的大小

** 让我们验证表的大小（之前执行innodb_stats_persistent_sample_pages=100000 的 ANALYZE，以便统计信息尽可能真实）。 **

```
set global innodb_stats_persistent_sample_pages=100000;
analyze table test_compress;

+-----------------------+---------+----------+----------+
| Table                 | Op      | Msg_type | Msg_text |
+-----------------------+---------+----------+----------+
| PS_57.test_compress   | analyze | status   | OK       |
+-----------------------+---------+----------+----------+
Query OK, 0 rows affected (0.00 sec)

select table_schema, table_name, table_rows,  round(data_length / 1024 / 1024)+round(index_
ength / 1024 / 1024)+round(data_free / 1024 / 1024) TOTAL_MB, create_options from inform
tion_schema.tables where table_name='test_compress';

+--------------+--------------+------------+----------+----------------+
| table_schema | table_name   | table_rows | TOTAL_MB | create_options |
+--------------+--------------+------------+----------+----------------+
| PS_57        | test_compress |     100000 |       37 |                |
+--------------+--------------+------------+----------+----------------+
```

3、对表压缩

**接下来，我们将用KEY_BLOCK_SIZE=4压缩表（这个大小是任意选择的，在任何时候都没有指示或定它是否是最优值，事实上，它不是）。 **

```
ALTER TABLE test_compress ROW_FORMAT=COMPRESSED,KEY_BLOCK_SIZE=4,ALGORITHM=
NPLACE,LOCK=NONE;

Query OK, 0 rows affected (3.33 sec)
```

** 我们再次验证表的大小（以前执行innodb_stats_persistent_sample_pages=100000 的 ANALYZE 表，以便统计信息尽可能真实）。 **

```
set global innodb_stats_persistent_sample_pages=100000;

Query OK, 0 rows affected (0.00 sec)

analyze table test_compress;

+----------------------+---------+----------+----------+
| Table                | Op      | Msg_type | Msg_text |
+----------------------+---------+----------+----------+
| PS_57.test_compress  | analyze | status   | OK       |
+----------------------+---------+----------+----------+
Query OK, 0 rows affected (0.00 sec)

select table_schema, table_name, table_rows,  round(data_length / 1024 / 1024)+round(index_
ength / 1024 / 1024)+round(data_free / 1024 / 1024) TOTAL_MB, create_options from inform
tion_schema.tables where table_name='test_compress';

+--------------+--------------+------------+----------+-----------------------------------+
| table_schema | table_name   | table_rows | TOTAL_MB | create_options                    |
+--------------+--------------+------------+----------+-----------------------------------+
| PS_57        | test_compress |     100000 |       19 | row_format=COMPRESSED KEY_BLOCK_SIZE
4 |
+--------------+--------------+------------+----------+-----------------------------------+
```

**该表已被压缩，让我们检查其结构。 **

```
show create table test_compress;
*************************** 1. row ***************************
       Table: test_compress
Create Table: CREATE TABLE `test_compress` (
  `id` bigint(20) unsigned NOT NULL,
  `identification_id` int(10) unsigned DEFAULT NULL,
  `timestamp` datetime NOT NULL,
  `action` varchar(50) NOT NULL,
  `result` varchar(50) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `INDEX_test_compress_result` (`result`),
  KEY `INDEX_test_compress_timestamp` (`timestamp`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 ROW_FORMAT=COMPRESSED KEY_BLOCK_SIZ
=4

1 row in set (0.00 sec)
```

4、压缩表解压缩（变成普通表）

```
ALTER TABLE test_compress ROW_FORMAT=DEFAULT,ALGORITHM=INPLACE,LOCK=NONE;

Query OK, 0 rows affected (6.25 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

解压缩成功，让我们检查看看。

```
select table_schema, table_name, table_rows,  round(data_length / 1024 / 1024)+round(index_
ength / 1024 / 1024)+round(data_free / 1024 / 1024) TOTAL_MB, create_options from inform
tion_schema.tables where table_name='test_compress';

+--------------+--------------+------------+----------+--------------------+
| table_schema | table_name    | table_rows | TOTAL_MB | create_options     |
+--------------+--------------+------------+----------+--------------------+
| PS_57        | test_compress |     100000 |       25 | KEY_BLOCK_SIZE=4   |
+--------------+--------------+------------+----------+--------------------+
```

** 更好的检查： **

```
show create table test_compress;

*************************** 1. row ***************************
       Table: test_compress
Create Table: CREATE TABLE `test_compress` (
  `id` bigint(20) unsigned NOT NULL,
  `identification_id` int(10) unsigned DEFAULT NULL,
  `timestamp` datetime NOT NULL,
  `action` varchar(50) NOT NULL,
  `result` varchar(50) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `INDEX_test_compress_result` (`result`),
  KEY `INDEX_test_compress_timestamp` (`timestamp`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 KEY_BLOCK_SIZE=4
```

** 出了点问题！KEY_BLOCK_SIZE仍然是4。 **

** 第二次尝试： **

```
ALTER TABLE test_compress ROW_FORMAT=DEFAULT,KEY_BLOCK_SIZE=0,ALGORITHM=INP
ACE,LOCK=NONE;

Query OK, 0 rows affected (2.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

select table_schema, table_name, table_rows,  round(data_length / 1024 / 1024)+round(index_
ength / 1024 / 1024)+round(data_free / 1024 / 1024) TOTAL_MB, create_options from inform
tion_schema.tables where table_name='test_compress';
+--------------+--------------+------------+----------+--------------------+
| table_schema | table_name    | table_rows | TOTAL_MB | create_options     |
+--------------+--------------+------------+----------+--------------------+
| PS_57        | test_compress |     100000 |       25 |                    |
+--------------+--------------+------------+----------+--------------------+
```

** 更好的检查： **

```
show create table test_compress\G
*************************** 1. row ***************************
       Table: test_compress
Create Table: CREATE TABLE `test_compress` (
  `id` bigint(20) unsigned NOT NULL,
  `identification_id` int(10) unsigned DEFAULT NULL,
  `timestamp` datetime NOT NULL,
```

```
  `action` varchar(50) NOT NULL,
  `result` varchar(50) NOT NULL,
  PRIMARY KEY (`id`) KEY_BLOCK_SIZE=4,
  KEY `INDEX_test_compress_result` (`result`) KEY_BLOCK_SIZE=4,
  KEY `INDEX_test_compress_timestamp` (`timestamp`) KEY_BLOCK_SIZE=4
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

出了点问题！主键和二级索引都继续显示 KEY_BLOCK_SIZE=4。

尽管当表从压缩转换为未压缩时，在内部，索引的KEY_BLOCK_SIZE支持表的索引，但 CREATE TABLE 语句则不然。起初，这将是一个美学/外观问题，但是当您进行转储时，这是一个真正的问题，因CREATE TABLE保留了KEY_BLOCK_SIZE值，这并不好。以下是 mysqldump 的输出：

```
mysqldump -h localhost PS_57 test_compress --no-data > test_compress.sql

cat test_compress.sql
...
--
-- Table structure for table `test_compress`
--

DROP TABLE IF EXISTS `test_compress`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `test_compress` (
  `id` bigint(20) unsigned NOT NULL,
  `identification_id` int(10) unsigned DEFAULT NULL,
  `timestamp` datetime NOT NULL,
  `action` varchar(50) NOT NULL,
  `result` varchar(50) NOT NULL,
  PRIMARY KEY (`id`) KEY_BLOCK_SIZE=4,
  KEY `INDEX_test_compress_result` (`result`) KEY_BLOCK_SIZE=4,
  KEY `INDEX_test_compress_timestamp` (`timestamp`) KEY_BLOCK_SIZE=4
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;
```

** 如您所见，似乎没有办法使用全局 ALTER TABLE 命令（如果可以这样称呼它）在表定义索引方面转KEY_BLOCK_SIZE，因此我们将进行最后一次尝试： **

```
ALTER TABLE test_compress
DROP PRIMARY KEY, add PRIMARY KEY (id),
DROP key INDEX_test_compress_result, add key INDEX_test_compress_result (result),
DROP key INDEX_test_compress_timestamp, add key INDEX_test_compress_timestamp (timestamp),
ROW_FORMAT=DEFAULT,KEY_BLOCK_SIZE=0,ALGORITHM=INPLACE,LOCK=NONE;
```

现在，它具有正确的定义，没有KEY_BLOCK_SIZE：

```
show create table test_compress;
*************************** 1. row ***************************
       Table: test_compress
Create Table: CREATE TABLE `test_compress` (
  `id` bigint(20) unsigned NOT NULL,
  `identification_id` int(10) unsigned DEFAULT NULL,
  `timestamp` datetime NOT NULL,
```

```
  `action` varchar(50) NOT NULL,
  `result` varchar(50) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `INDEX_test_compress_result` (`result`),
  KEY `INDEX_test_compress_timestamp` (`timestamp`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
1 row in set (0.00 sec)

select table_schema, table_name, table_rows,  round(data_length / 1024 / 1024)+round(index_
ength / 1024 / 1024)+round(data_free / 1024 / 1024) TOTAL_MB, create_options from inform
tion_schema.tables where table_name='test_compress';
+--------------+--------------+------------+----------+----------------+
| table_schema | table_name    | table_rows | TOTAL_MB | create_options |
+--------------+--------------+------------+----------+----------------+
| PS_57        | test_compress |     100000 |       25 |                |
+--------------+--------------+------------+----------+----------------+
```

5、针对第4步出现问题的bug

mysql里有解释这个bug: **https://bugs.mysql.com/bug.php?id=56628 **

针对mysql8实验

** 在MySQL 8中，情况如下： **

```
select table_schema, table_name, table_rows,  round(data_length / 1024 / 1024)+round(index_
ength / 1024 / 1024)+round(data_free / 1024 / 1024) TOTAL_MB, create_options from inform
tion_schema.tables where table_name='test_compress';

+--------------+--------------+------------+----------+----------------+
| TABLE_SCHEMA | TABLE_NAME    | TABLE_ROWS | TOTAL_MB | CREATE_OPTIONS |
+--------------+--------------+------------+----------+----------------+
| PS_8         | test_compress |      31000 |       15 |                |
+--------------+--------------+------------+----------+----------------+
```

** 让我们执行 ALTER 来压缩表： **

```
alter table test_compress ROW_FORMAT=COMPRESSED,KEY_BLOCK_SIZE=4,ALGORITHM=IN
LACE,LOCK=NONE;

Query OK, 0 rows affected (4.54 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

** 让我们再检查一下： **

```
analyze table test_compress;

+----------------------+---------+----------+----------+
| Table                | Op      | Msg_type | Msg_text |
+----------------------+---------+----------+----------+
| PS_8.test_compress   | analyze | status   | OK       |
+----------------------+---------+----------+----------+
1 row in set (0.07 sec)
```

```
select table_schema, table_name, table_rows,  round(data_length / 1024 / 1024)+round(index_
ength / 1024 / 1024)+round(data_free / 1024 / 1024) TOTAL_MB, create_options from inform
tion_schema.tables where table_name='test_compress';

+--------------+--------------+------------+----------+-----------------------------------+
| TABLE_SCHEMA | TABLE_NAME    | TABLE_ROWS | TOTAL_MB | CREATE_OPTIONS
    |
+--------------+--------------+------------+----------+-----------------------------------+
| PS_8         | test_compress |     100000 |       19 | row_format=COMPRESSED KEY_BLOCK_SIZE
4 |
+--------------+--------------+------------+----------+-----------------------------------+

show create table test_compress;
*************************** 1. row ***************************
       Table: test_compress
Create Table: CREATE TABLE `test_compress` (
  `id` bigint unsigned NOT NULL,
  `identification_id` int unsigned DEFAULT NULL,
  `timestamp` datetime NOT NULL,
 `action` varchar(50) NOT NULL,
  `result` varchar(50) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `INDEX_test_compress_result` (`result`),
  KEY `INDEX_test_compress_timestamp` (`timestamp`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 ROW_FORMAT=COMPRESSED KEY_BLOCK_SIZ
=4
1 row in set (0.01 sec)
```

** 到目前为止，一切都与MySQL 5.7相同：KEY_BLOCK_SIZE保留在整个表的定义中，而不是索引的义中。 **

同样的，也能通过下面sql对表进行解压缩：

```
alter table test_compress ROW_FORMAT=DEFAULT, KEY_BLOCK_SIZE=0,ALGORITHM=INPLA
E,LOCK=NONE;

Query OK, 0 rows affected (2.56 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

查看解压缩情况

```
show create table test_compress;

*************************** 1. row ***************************
       Table: test_compress
Create Table: CREATE TABLE `test_compress` (
  `id` bigint unsigned NOT NULL,
  `identification_id` int unsigned DEFAULT NULL,
  `timestamp` datetime NOT NULL,
  `action` varchar(50) NOT NULL,
  `result` varchar(50) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `INDEX_test_compress_result` (`result`),
  KEY `INDEX_test_compress_timestamp` (`timestamp`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

```
1 row in set (0.00 sec)

select table_schema, table_name, table_rows,  round(data_length / 1024 / 1024)+round(index_
ength / 1024 / 1024)+round(data_free / 1024 / 1024) TOTAL_MB, create_options from inform
tion_schema.tables where table_name='test_compress';

+--------------+---------------+------------+----------+----------------+
| TABLE_SCHEMA | TABLE_NAME    | TABLE_ROWS | TOTAL_MB | CREATE_OPTIONS |
+--------------+---------------+------------+----------+----------------+
| PS_8         | test_compress |     100000 |       25 |                |
+--------------+---------------+------------+----------+----------------+
```

结论

** 在MySQL 5.7中，完全解压缩一张压缩表的唯一方法（至少在表及其索引的定义中）是重新生成主及其所有索引。否则， 主键和二级索引都继续显示压缩表时候的KEY_BLOCK_SIZE。**

然后在MySQL8里，修复了这个问题在MySQL5.7出现的问题。

本文原创作者：奇想派、一名努力分享的程序员。

文章首发平台：微信公众号【编程达人】

原创不易！各位小伙伴觉得文章不错的话，不妨关注公众号，进行**点赞（在看）、转发**三连走起！谢大家！