

优化内容： 通话记录表called_records

说明： 测试都是在同一台机器的环境

原始数据：

```
***** 1. row *****
      Name: called_records_origin
      Engine: InnoDB
      Version: 10
      Row_format: Dynamic
      Rows: 3726330
      Avg_row_length: 1044
      Data_length: 3892314112
      Max_data_length: 0
      Index_length: 1086111744
      Data_free: 4194304
      Auto_increment: 3873827
      Create_time: 2017-11-02 10:12:44
      Update_time: NULL
      Check_time: NULL
      Collation: utf8_unicode_ci
      Checksum: NULL
      Create_options:
      Comment:
1 row in set (0.00 sec)
```

慢日志记录sql测试时间（所有查询都不走缓存）：

SQL: select sql_no_cache `id`,`call_sheet_id`,`called_state`,`call_id`,`record_file` from `called_records_origin` where `created_at` > '2017-11-02 11:30:01';
Time: 12.23s
Explain:

```
***** 1. row *****
      id: 1
      select_type: SIMPLE
      table: called_records_origin
      partitions: NULL
      type: ALL
possible_keys: NULL
  key: NULL
  key_len: NULL
  ref: NULL
  rows: 3726330
  filtered: 33.33
  Extra: Using where
1 row in set, 1 warning (0.01 sec)
```

SQL: select sql_no_cache count(id) as times from `called_records_origin` where `exten` = '8038' and `call_type` = 'dialout' and `state_code` <> '110' and `created_at` >= '2017-11-02 00:00:00' limit 1;
Time: 7.34s
Explain:

```
***** 1. row *****
      id: 1
      select_type: SIMPLE
      table: called_records_origin
      partitions: NULL
      type: ref
possible_keys: called_records_exten_index
  key: called_records_exten_index
  key_len: 15
  ref: const
  rows: 62528
  filtered: 3.00
  Extra: Using where
1 row in set, 1 warning (0.01 sec)
```

SQL: select sql_no_cache count(*) as aggregate from `called_records_origin` where `state_code` <> '110' and `sale_name` in ('张晨雨') and `call_time_length` >= '0' and `call_time_length` <= '99999' and `sale_uid` in ('689') order by `created_at` desc;
Time: 4.49s
Explain:

```
***** 1. row *****
      id: 1
      select_type: SIMPLE
      table: called_records_origin
      partitions: NULL
      type: index_merge
possible_keys: called_records_sale_uid_index,called_records_sale_uid_clue_id_index,called_records_sale_name_index
  key: called_records_sale_name_index,called_records_sale_uid_index
  key_len: 152,767
  ref: NULL
  rows: 375
  filtered: 10.00
  Extra: Using intersect(called_records_sale_name_index,called_records_sale_uid_index); Using where
1 row in set, 1 warning (0.01 sec)
```

根据创建时间分区后：

```
mysql> show table status like "called_records_part"\G;
***** 1. row *****
      Name: called_records_part
      Engine: InnoDB
      Version: 10
      Row_format: Dynamic
      Rows: 0
      Avg_row_length: 0
      Data_length: 245760
      Max_data_length: 0
      Index_length: 2211840
      Data_free: 45088768
      Auto_increment: 3873827
      Create_time: 2017-11-02 10:48:15
      Update_time: 2017-11-02 10:45:34
      Check_time: NULL
      Collation: utf8_unicode_ci
      Checksum: NULL
      Create_options: partitioned
      Comment:
1 row in set (0.00 sec)
```

慢日志记录sql测试时间（所有查询都不走缓存）：

SQL: select sql_no_cache `id`,`call_sheet_id`,`called_state`,`call_id`,`record_file` from `called_records_part` where `created_at` > '2017-11-02 11:30:01';
Time: 0.00s（此数据稍有特殊）
Explain:

```
***** 1. row *****
      id: 1
      select_type: SIMPLE
      table: called_records_part
      partitions: p201711,p201712,p201801,p201802,p201803,p201804,p201805,p201806,p999999
      type: ALL
possible_keys: NULL
  key: NULL
  key_len: NULL
  ref: NULL
  rows: 1
  filtered: 100.00
  Extra: Using where
1 row in set, 1 warning (0.00 sec)
```

SQL: select sql_no_cache count(id) as times from `called_records_part` where `exten` = '8038' and `call_type` = 'dialout' and `state_code` <> '110' and `created_at` >= '2017-11-02 00:00:00' limit 1;
Time: 0.01s
Explain:

```
***** 1. row *****
      id: 1
      select_type: SIMPLE
      table: called_records_part
      partitions: p201711,p201712,p201801,p201802,p201803,p201804,p201805,p201806,p999999
      type: ref
possible_keys: called_records_exten_index
  key: called_records_exten_index
  key_len: 15
  ref: const
  rows: 1
  filtered: 100.00
  Extra: Using index condition; Using where
1 row in set, 1 warning (0.00 sec)
```

SQL: select sql_no_cache count(*) as aggregate from `called_records_part` where `state_code` <> '110' and `sale_name` in ('张晨雨') and `call_time_length` >= '0' and `call_time_length` <= '99999' and `sale_uid` in ('689') order by `created_at` desc;
Time: 5.22s
Explain:

```
***** 1. row *****
      id: 1
      select_type: SIMPLE
      table: called_records_part
      partitions: p201705,p201706,p201707,p201708,p201709,p201710,p201711,p201712,p201801,p201802,p201803,p201804,p201805,p201806,p999999
      type: ref
possible_keys: called_records_sale_uid_index,called_records_sale_uid_clue_id_index,called_records_sale_name_index
  key: called_records_sale_uid_clue_id_index
  key_len: 767
  ref: const
  rows: 1
  filtered: 100.00
  Extra: Using where
1 row in set, 1 warning (0.01 sec)
```

字段瘦身：

```
mysql> show table status like "called_records_small"\G;
***** 1. row *****
      Name: called_records_small
      Engine: InnoDB
      Version: 10
      Row_format: Dynamic
      Rows: 3791525
      Avg_row_length: 301
      Data_length: 1142931456
      Max_data_length: 0
      Index_length: 1064779776
      Data_free: 5242880
      Auto_increment: 3873827
      Create_time: 2017-11-02 15:42:08
      Update_time: NULL
      Check_time: NULL
      Collation: utf8_unicode_ci
      Checksum: NULL
      Create_options:
      Comment:
1 row in set (0.00 sec)
```

慢日志记录sql测试时间（所有查询都不走缓存）：

SQL: select sql_no_cache `id`,`call_sheet_id`,`called_state`,`call_id`,`record_file` from `called_records_small` where `created_at` > '2017-11-02 11:30:01';
Time: 3.51s
Explain:

```
***** 1. row *****
      id: 1
      select_type: SIMPLE
      table: called_records_small
      partitions: NULL
      type: ALL
possible_keys: NULL
  key: NULL
  key_len: NULL
  ref: NULL
  rows: 3791525
  filtered: 33.33
  Extra: Using where
1 row in set, 1 warning (0.00 sec)
```

SQL: select sql_no_cache count(id) as times from `called_records_small` where `exten` = '8038' and `call_type` = 'dialout' and `state_code` <> '110' and `created_at` >= '2017-11-02 00:00:00' limit 1;
Time: 0.01s
Explain:

```
***** 1. row *****
      id: 1
      select_type: SIMPLE
      table: called_records_small
      partitions: NULL
      type: ref
possible_keys: called_records_exten_index
  key: called_records_exten_index
  key_len: 2
  ref: const
  rows: 62906
  filtered: 3.00
  Extra: Using index condition; Using where
1 row in set, 1 warning (0.00 sec)
```

SQL: select sql_no_cache count(*) as aggregate from `called_records_small` where `state_code` <> '110' and `sale_name` in ('张晨雨') and `call_time_length` >= '0' and `call_time_length` <= '99999' and `sale_uid` in ('689') order by `created_at` desc;
Time: 0.57s
Explain:

```
***** 1. row *****
      id: 1
      select_type: SIMPLE
      table: called_records_small
      partitions: NULL
      type: index_merge
possible_keys: called_records_sale_uid_index,called_records_sale_uid_clue_id_index,called_records_sale_name_index
  key: called_records_sale_uid_index,called_records_sale_name_index
  key_len: 4,92
  ref: NULL
  rows: 468
  filtered: 10.00
  Extra: Using intersect(called_records_sale_uid_index,called_records_sale_name_index); Using where
1 row in set, 1 warning (0.00 sec)
```

字段瘦身&分区后：

```
***** 1. row *****
      Name: called_records_part_small
      Engine: InnoDB
      Version: 10
      Row_format: Dynamic
      Rows: 3826630
      Avg_row_length: 299
      Data_length: 1146683392
      Max_data_length: 0
      Index_length: 1084391424
      Data_free: 19922944
      Auto_increment: 3873827
      Create_time: 2017-11-02 16:21:12
      Update_time: NULL
      Check_time: NULL
      Collation: utf8_unicode_ci
      Checksum: NULL
      Create_options: partitioned
      Comment:
1 row in set (0.01 sec)
```

慢日志记录sql测试时间（所有查询都不走缓存）：

SQL: select sql_no_cache `id`,`call_sheet_id`,`called_state`,`call_id`,`record_file` from `called_records_part_small` where `created_at` > '2017-11-02 11:30:01';
Time: 0.00s
Explain:

```
***** 1. row *****
      id: 1
      select_type: SIMPLE
      table: called_records_part_small
      partitions: p201711,p201712,p201801,p201802,p201803,p201804,p201805,p201806,p999999
      type: ALL
possible_keys: NULL
  key: NULL
  key_len: NULL
  ref: NULL
  rows: 530
  filtered: 33.33
  Extra: Using where
1 row in set, 1 warning (0.00 sec)
```

SQL: select sql_no_cache count(id) as times from `called_records_part_small` where `exten` = '8038' and `call_type` = 'dialout' and `state_code` <> '110' and `created_at` >= '2017-11-02 00:00:00' limit 1;
Time: 0.01s
Explain:

```
***** 1. row *****
      id: 1
      select_type: SIMPLE
      table: called_records_part_small
      partitions: p201711,p201712,p201801,p201802,p201803,p201804,p201805,p201806,p999999
      type: ref
possible_keys: called_records_exten_index
  key: called_records_exten_index
  key_len: 2
  ref: const
  rows: 1
  filtered: 5.00
  Extra: Using index condition; Using where
1 row in set, 1 warning (0.00 sec)
```

SQL: select sql_no_cache count(*) as aggregate from `called_records_part_small` where `state_code` <> '110' and `sale_name` in ('张晨雨') and `call_time_length` >= '0' and `call_time_length` <= '99999' and `sale_uid` in ('689') order by `created_at` desc;
Time: 1.05s
Explain:

```
***** 1. row *****
      id: 1
      select_type: SIMPLE
      table: called_records_part_small
      partitions: p201705,p201706,p201707,p201708,p201709,p201710,p201711,p201712,p201801,p201802,p201803,p201804,p201805,p201806,p999999
      type: index_merge
possible_keys: called_records_sale_uid_index,called_records_sale_uid_clue_id_index,called_records_sale_name_index
  key: called_records_sale_uid_index,called_records_sale_name_index
  key_len: 4,92
  ref: NULL
  rows: 122
  filtered: 10.00
  Extra: Using intersect(called_records_sale_uid_index,called_records_sale_name_index); Using where
1 row in set, 1 warning (0.00 sec)
```

针对上述特殊情况做特殊对比：
SQL: select sql_no_cache count(*) as aggregate from `called_records_part_small` where `state_code` <> '110' and `sale_name` in ('张晨雨') and `call_time_length` >= '0' and `call_time_length` <= '99999' and `sale_uid` in ('689') order by `created_at` desc;
Time: 0.06s
SQL: select sql_no_cache count(*) as aggregate from `called_records_small` where `state_code` <> '110' and `sale_name` in ('张晨雨') and `call_time_length` >= '0' and `call_time_length` <= '99999' and `sale_uid` in ('689') order by `created_at` desc;
Time: 0.45s

SQL: select sql_no_cache count(*) as aggregate from `called_records_part` where `state_code` <> '110' and `sale_name` in ('张晨雨') and `call_time_length` >= '0' and `call_time_length` <= '99999' and `sale_uid` in ('689') order by `created_at` desc;
Time: 1.20s
SQL: select sql_no_cache count(*) as aggregate from `called_records_origin` where `state_code` <> '110' and `sale_name` in ('张晨雨') and `call_time_length` >= '0' and `call_time_length` <= '99999' and `sale_uid` in ('689') order by `created_at` desc;
Time: 7.10s

```
mysql> select sql_no_cache count(id) as times from called_records_part_small where `exten` = '8038' and `call_type` = 'dialout' and `state_code` <> '110' and `created_at` >= '2017-10-02 00:00:00' limit 1;
+-----+
| times |
+-----+
| 8064 |
+-----+
1 row in set (0.00 sec)

mysql> select sql_no_cache count(id) as times from called_records_small where `exten` = '8038' and `call_type` = 'dialout' and `state_code` <> '110' and `created_at` >= '2017-10-02 00:00:00' limit 1;
+-----+
| times |
+-----+
| 8064 |
+-----+
1 row in set (0.45 sec)

mysql> select sql_no_cache count(id) as times from called_records_part where `exten` = '8038' and `call_type` = 'dialout' and `state_code` <> '110' and `created_at` >= '2017-10-02 00:00:00' limit 1;
+-----+
| times |
+-----+
| 8064 |
+-----+
1 row in set (1.20 sec)

mysql> select sql_no_cache count(id) as times from called_records_origin where `exten` = '8038' and `call_type` = 'dialout' and `state_code` <> '110' and `created_at` >= '2017-10-02 00:00:00' limit 1;
+-----+
| times |
+-----+
| 8064 |
+-----+
1 row in set (7.10 sec)
```

备注：
表的瘦身还可以再做，将字典类型的字段改成int类(int,smallint,tinyint等)类型。但是这个目前对业务影响较大，暂不实施。