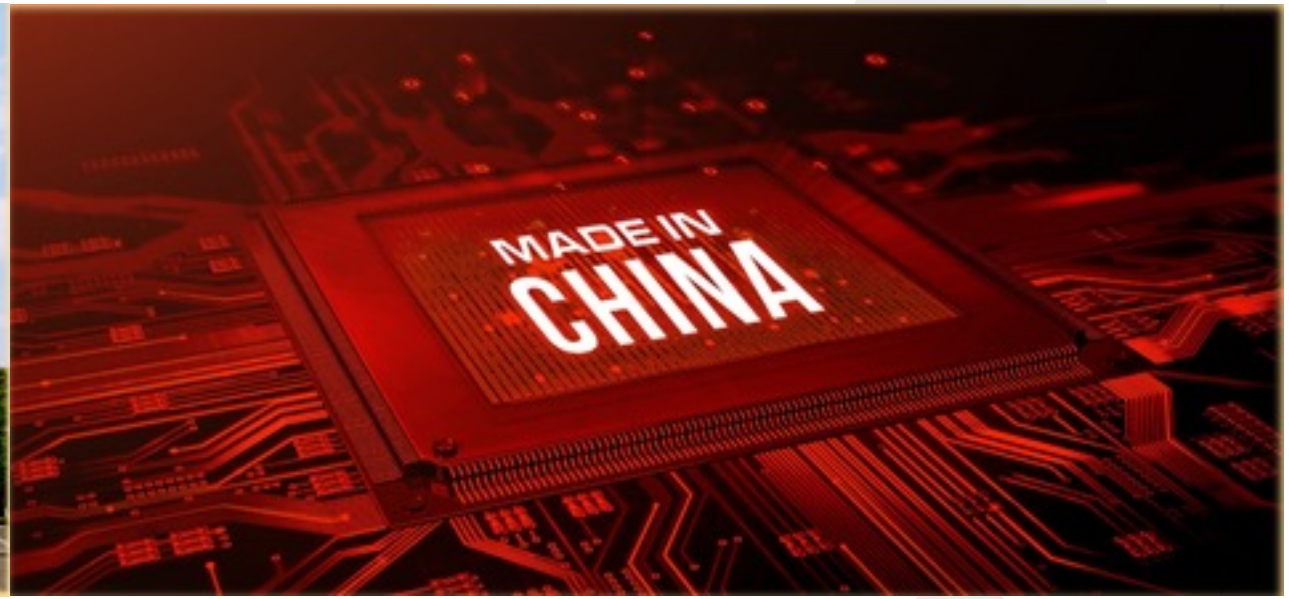
GreatDB
万里数据库



闪回在MySQL中的实现和改进

唐洁

「3306π」福州站,2022.1.15

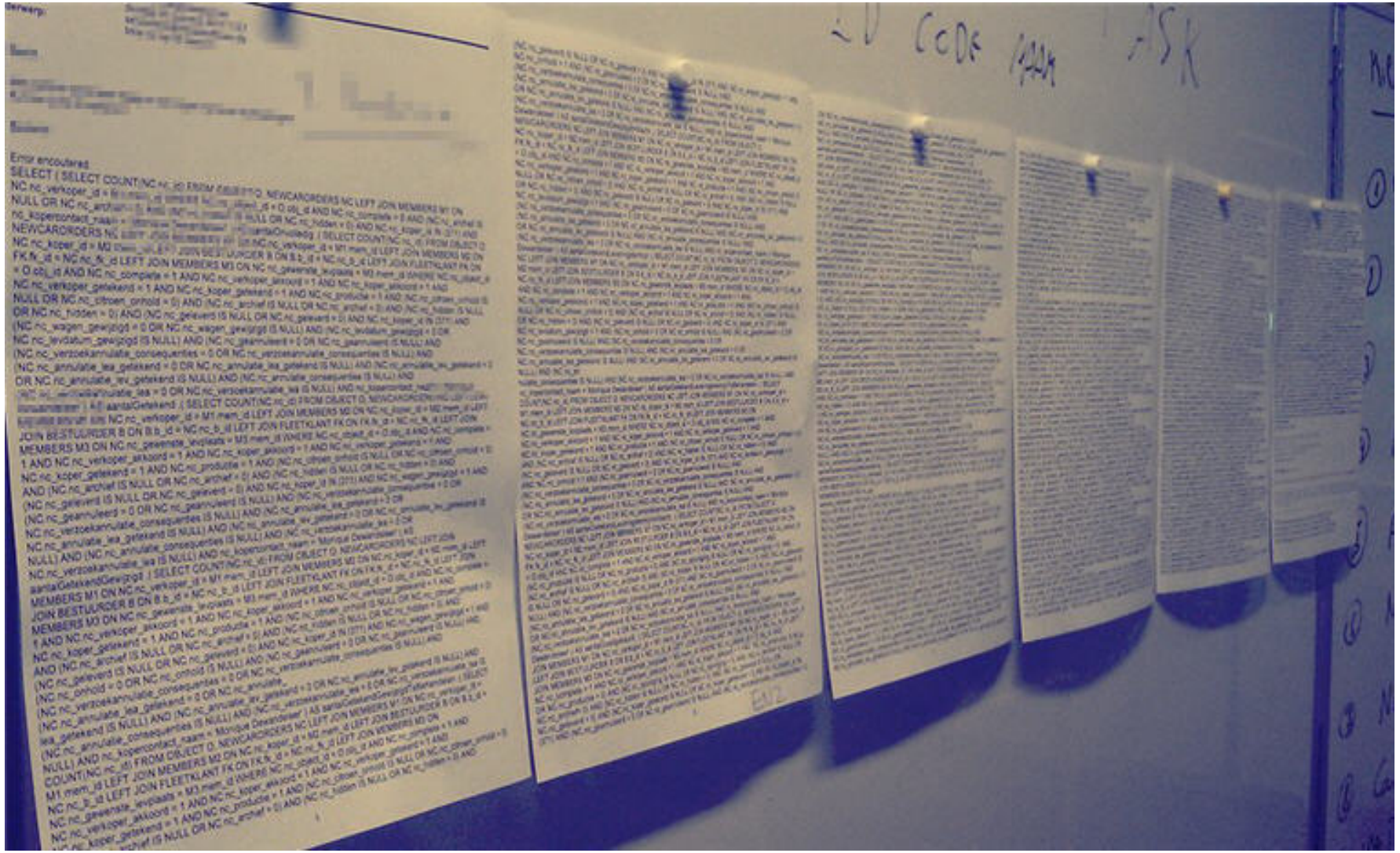
GreatDB
万里数据库

目录

CONTENTS

绝望的SQL

GreatDB
万里数据库



数据库闪回能力的业务价值

GreatDB 万里数据库

人祸所为

- **沟通**：IT流程的环节由不同的人员各司其职分工完成。文档？口头？传递信息的缺失，沟通不畅导致
- **失误**：DBA手滑写错/眼花看错，生而为人难以避免
- **工期**：客户催老板急，开发人员赶工完成紧急任务，数据脚本未进行充分测试

价值意义

- **便捷**：数据备份无需DBA专门操作，而是由数据库自动触发
- **补救**：在DBA执行错误的数据提交操作之后，还能把数据恢复还原到之前某个时刻的状态，最大程度的挽回损失

binlog文件功能

GreatDB 万里数据库

binlog文件用途

- **mysql主服务器同步触发：**同步二进制数据日志给从服务器：主服务器数据发生更新后，会把变动，以event的形式，记录binlog文件，然后从服务器会拉取binlog文件解析。从服务器通过拉取并解析binlog文件，实现数据的同步。
- **记录操作轨迹：**binlog文件中记录了数据变更的信息。
- **数据恢复：**借助binlog日志，可以恢复到故障之前的形态。

binlog文件更新方式

- **Statement：**文件中存的是sql语句，优点是传输的数据量比较少，缺点是很难保证主从一致。
- **Row：**文件中存的是更新的那一行的数据内容，优点是不会出错，缺点是传输的数据量比较大。
- **Mixed：**Statement和Row模式的结合。

配置binlog

GreatDB 万里数据库

配置binlog文件

- 查看log_bin系统变量来判断当前MySQL服务器是否生成binlog日志

```
mysql> show variables like 'log_bin';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| log_bin       | ON    |
+---------------+-------+
1 row in set, 1 warning (0.02 sec)
```

- 上例中bin_log系统变量的值为ON，表明当前服务器生成binlog，若为OFF表明当前服务器不生成binlog。

启用binlog文件

- 重启服务器，设置log-bin启动选项：

  --log-bin[=file_name]

  例如：mysqld --log-bin=/user/greatdb/test

- log-bin启动选项也可以放在cnf配置文件

生成binlog文件

- 表示开启binlog，并将binlog写入MySQL服务器的数据目录/user/greatdb/下，binlog日志文件名就像是这样：

  test.000001

  test.000002

  ……

binlog文件格式

GreatDB 万里数据库

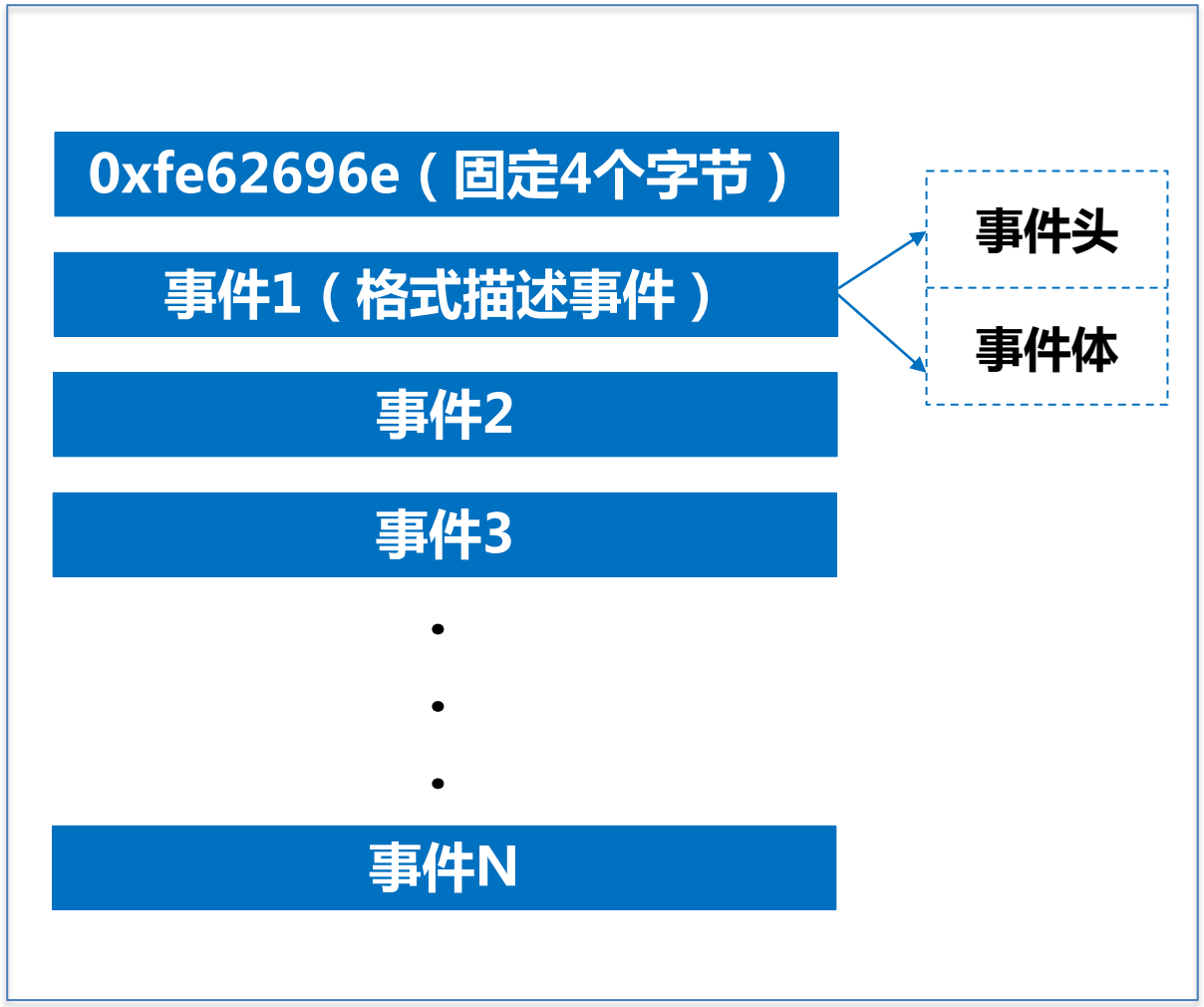
binlog文件的位置

- show variables like '%datadir% ‘
- hexdump -C binlog.000006|more

输出

```
00000000  fe 62 69 6e ad 10 15 61  0f 01 00 00 00 79 00 00  |.bin...a.....y..|
00000010  00 7d 00 00 00 00 00 04  00 38 2e 30 2e 32 35 2d  |.}.......8.0.25-|
00000020  31 35 2d 64 65 62 75 67  00 00 00 00 00 00 00 00  |15-debug........|
00000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000040  00 00 00 00 00 00 00 00  00 00 00 ad 10 15 61 13  |..............a.|
00000050  00 0d 00 08 00 00 00 00  04 00 04 00 00 00 61 00  |..............a.|
00000060  04 1a 08 00 00 00 08 08  08 02 00 00 00 0a 0a 0a  |................|
00000070  2a 2a 00 12 34 00 0a 28  01 ea d7 cf 01 ad 10 15  |**..4..(........|
00000080  61 23 01 00 00 00 1f 00  00 00 9c 00 00 00 80 00  |a#..............|
```

- 前面的4个字节**fe 62 69 6e**是魔数，标识文件类型是binlog。
- 后面的二进制数据则表示事件详细数据。



binlog文件组成

binlog文件结构

GreatDB 万里数据库

主要事件类型

- FORMAT_DESCRIPTION_EVENT
- ANONYMOUS_GTID_LOG_EVENT/GTID_LOG_EVENT
- QUERY_EVENT
- TABLE_MAP_EVENT
- WRITE_ROWS_EVENT/UPDATE_ROWS_EVENT/DELETE_ROWS_EVENT
- XID_EVENT

事件组成

- 事件头+事件体

事件要素查看

- 客户端：SHOW BINLOG EVENTS [IN 'log_name'] [FROM pos] [LIMIT [offset,] row_count]
- 命令行：mysqlbinlog

客户端方式查看binlog事件

GreatDB 万里数据库

mysql客户端执行SHOW BINLOG EVENTS，表示查看第1个binlog日志文件的内容。

```
mysql> SHOW BINLOG EVENTS;
+----------------------+-----+----------------+-----------+-------------+-------------------------------------+
| Log_name             | Pos | Event_type     | Server_id | End_log_pos | Info                                |
+----------------------+-----+----------------+-----------+-------------+-------------------------------------+
| greatdb-bin.000001   |   4 | Format_desc    |         3 |         123 | Server ver: 5.7.21-log, Binlog ver: 4 |
| greatdb-bin.000001   | 123 | Previous_gtids |         3 |         154 |                                     |
| greatdb-bin.000001   | 154 | Anonymous_Gtid |         3 |         219 | SET @@SESSION.GTID_NEXT= 'ANONYMOUS'  |
| greatdb-bin.000001   | 219 | Query          |         3 |         296 | BEGIN                               |
| greatdb-bin.000001   | 296 | Table_map      |         3 |         367 | table_id: 138 (users.s1)            |
| greatdb-bin.000001   | 367 | Update_rows    |         3 |         634 | table_id: 138 flags: STMT_END_F     |
| greatdb-bin.000001   | 634 | Xid            |         3 |         665 | COMMIT /* xid=65 */                 |
| greatdb-bin.000001   | 665 | Stop           |         3 |         688 |                                     |
+----------------------+-----+----------------+-----------+-------------+-------------------------------------+
```

主要列说明

- Log_name
- Pos
- Event_type
- Server_id
- End_log_pos
- Info

命令行方式查看binlog事件

GreatDB 万里数据库

查看测试数据对应的binlog文件内容：

mysqlbinlog --base64-output=decode-rows -vv /home/percona/percona/release/data/binlog.000031

```
# at 2014 ====>起始位置2014
#210727 14:19:27 server id 1  end_log_pos 2093 CRC32 0x0a059009      1  Anonymous_GTID  ......
# at 2093
#210727 14:19:27 server id 1  end_log_pos 2167 CRC32 0xf3f13bd4      2  Query   thread_id=8     exec_time=0     error_code=0
# at 2167
#210727 14:19:27 server id 1  end_log_pos 2235 CRC32 0x2cb803c9      3  Table_map: `ccs`.`flashback_test` mapped to number 112
# at 2235
#210727 14:19:27 server id 1  end_log_pos 2302 CRC32 0xce5c009b      4  Write_rows: table id 112 flags: STMT_END_F
### INSERT INTO `ccs`.`flashback_test`
### SET
###   @1=4 /* INT meta=0 nullable=0 is_null=0 */
###   @2=40001 /* LONGINT meta=0 nullable=0 is_null=0 */
###   @3='测试数据40001' /* VARSTRING(1024) meta=1024 nullable=1 is_null=0 */
# at 2302
#210727 14:19:27 server id 1  end_log_pos 2333 CRC32 0xbef3fa89      5  Xid = 41
# at 2333
#210727 14:19:27 server id 1  end_log_pos 2412 CRC32 0x4739878e          Anonymous_GTID  ......
# at 2412
#210727 14:19:27 server id 1  end_log_pos 2495 CRC32 0xb4f01b33          Query   thread_id=8     exec_time=0     error_code=0
# at 2495
#210727 14:19:27 server id 1  end_log_pos 2563 CRC32 0x8955b862          Table_map: `ccs`.`flashback_test` mapped to number 112
# at 2563
#210727 14:19:27 server id 1  end_log_pos 2639 CRC32 0x132dc148          Update_rows: table id 112 flags: STMT_END_F
### UPDATE `ccs`.`flashback_test`
### WHERE
###   @1=2 /* INT meta=0 nullable=0 is_null=0 */
###   @2=20001 /* LONGINT meta=0 nullable=0 is_null=0 */
###   @3=NULL /* VARSTRING(1024) meta=1024 nullable=1 is_null=1 */
### SET
###   @1=2 /* INT meta=0 nullable=0 is_null=0 */
```

要素构成

binlog中的最小事务单元的构成

- Anonymous_Gtid/Gtid
- Query
- Table_map
- Write_rows/Delete_rows/Update_rows
- Xid

GreatDB
万里数据库

目录

CONTENTS

竞品分析（1/4）：oracle/ob

GreatDB 万里数据库

实现原理

- 基于回收站机制，把数据库改动前的镜像放到undo表空间中。
- 如果要回滚某个数据库对象, 只需要找到undo表空间中对应的undo数据即可。

优点

- 支持数据库级别（当误删除用户模式时适用）。
- 支持表级（drop/truncate命令删除的表适用）。
- 支持事务级闪回（对数据的增删改操作适用，只支持时间戳方式检索）。

缺点

- 闪回一般只适用于短时间内的数据恢复。
- 只能根据时间戳方式恢复。

事务级闪回操作步骤样例（SQL模式）

- 根据时间戳查询待闪回的历史记录：select * from 表名 as of timestamp to_timestamp('2021-08-04 11:00:00','yyyy-mm-dd hh24:mi:ss');
- 数据回滚闪回： flashback table 表名 to timestamp to_timestamp('2021-08-04 11:00:00','yyyy-mm-dd hh24:mi:ss');

竞品分析（2/4）：binlog2sql

GreatDB 万里数据库

实现原理

- 伪装成slave拉取binlog，从MySQL binlog解析出具体的SQL操作。
- 对其进行逆操作，最后在反向输出。

优点

- 简单python脚本，修改安装简单，支持增删改操作的回滚。

缺点

- mysql服务器必须开启，不支持离线模式。
- 不支持DDL语句，解析速度慢，效率低。

操作步骤样例（命令行模式）

- 生成闪回sql脚本：binlog2sql.py -h127.0.0.1 -P3306 -uadmin -p'admin' -dtest -tuser --start-file='mysql-bin.000054' --start-datetime='2021-08-01 11:00:00' --stop-datetime='2021-08-04 11:00:00' > rollback.sql
- 闪回脚本导入：mysql -h127.0.0.1 -P3306 -uadmin -p'admin' < rollback.sql

竞品分析（3/4）：AliSQL数据库闪回

GreatDB 万里数据库

实现原理

- 基于MySQL，对于数据增删改的操作，mysql中的binlog日志以事件的形式，记录了数据操作的前后差异。
- 解析binlog中的事件，然后反序遍历，同时对增删改进行逆操作，最后输出对应数据回滚的binlog，将其导入mysql即完成数据的闪回。

优点

- 在MySQL自带的mysqlbinlog命令中新增闪回参数-B，无需额外安装命令工具。

缺点

- 不支持DDL语句的回滚，升级困难。实现方式和MySQL内核代码紧密耦合，需要对mysql源码编译打补丁。

操作步骤样例（命令行模式）

- 生成闪回binlog文件：mysqlbinlog -B -v --start-position=8602 --stop-position=8783 /home/mysql/binlog.000013 > rollback.out
- 数据回滚闪回：mysqlbinlog rollback.out | mysql -u percona -ppercona -h127.0.0.1

竞品分析（4/4）：TiDB闪回

GreatDB 万里数据库

实现原理

- TIDB内部也实现了类似MySQL的binlog文件，因此实现原理同MySQL闪回模式。

优点

- 自带的drainer命令新增时间戳：-start-time、-end-time参数实现。

缺点

- 不支持DDL语句的回滚，只支持时间戳方式闪回。

操作步骤样例（命令行模式）

- 生成闪回binlog文件：输入drainer -pd-urls https://127.0.0.1:2379 -dest-type flashback -start-time "209-12-25 00:00:00" -end-time "2019-12-25 10:00:00"，生成闪回的binlog文件
- 数据回滚闪回：reparo -data-dir binlog文件目录

GreatDB
万里数据库

目录

CONTENTS

DML闪回实现方案

GreatDB 万里数据库

实现原理

- 基于binlog的实现方式：解析并处理binlog文件中的事件，然后反序遍历，同时对增删改进行反转逆操作。

实现功能

- 新增闪回命令flashback方式实现。
- 支持DML语句闪回。

操作步骤样例（命令行模式）

- 生成闪回binlog文件：

  flashback --start-position=8602

  --stop-position=8783

  --binlogFileNames=/home/mysql/binlog.000013 > rollback.out

- 数据回滚闪回：

  mysqlbinlog rollback.out | mysql -u percona -ppercona

MySQL字段长度限制的计算方法

packlen：一种可变长的存储数值的协议

- 如果第一个字节数值小于0xfb（251），则该数值通过1个字节存储；
- 如果第一个字节数值等于0xfc（252），则该数值通过2个字节存储；
- 如果第一个字节数值等于0xfd（253），则该数值通过3个字节存储；
- 如果第一个字节数值等于0xfe（254），则该数值通过8个字节存储；

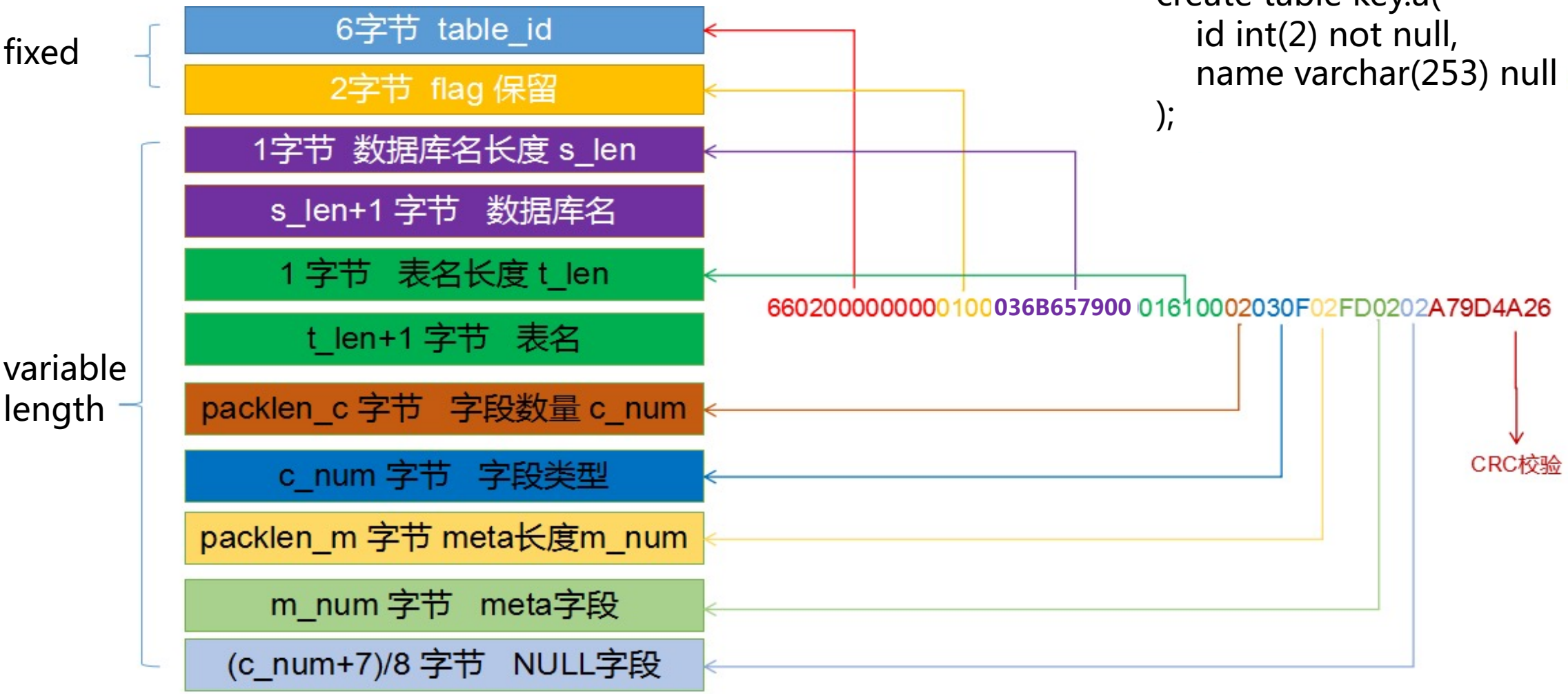
packlen的优点

- 读取第一个字节，根据第一个字节的值，来获取数据真正有多少个字节。
- 节约存储空间。

闪回binlog事件解析过程（1/3）

GreatDB 万里数据库

TABLE_MAP_EVENT事件结构



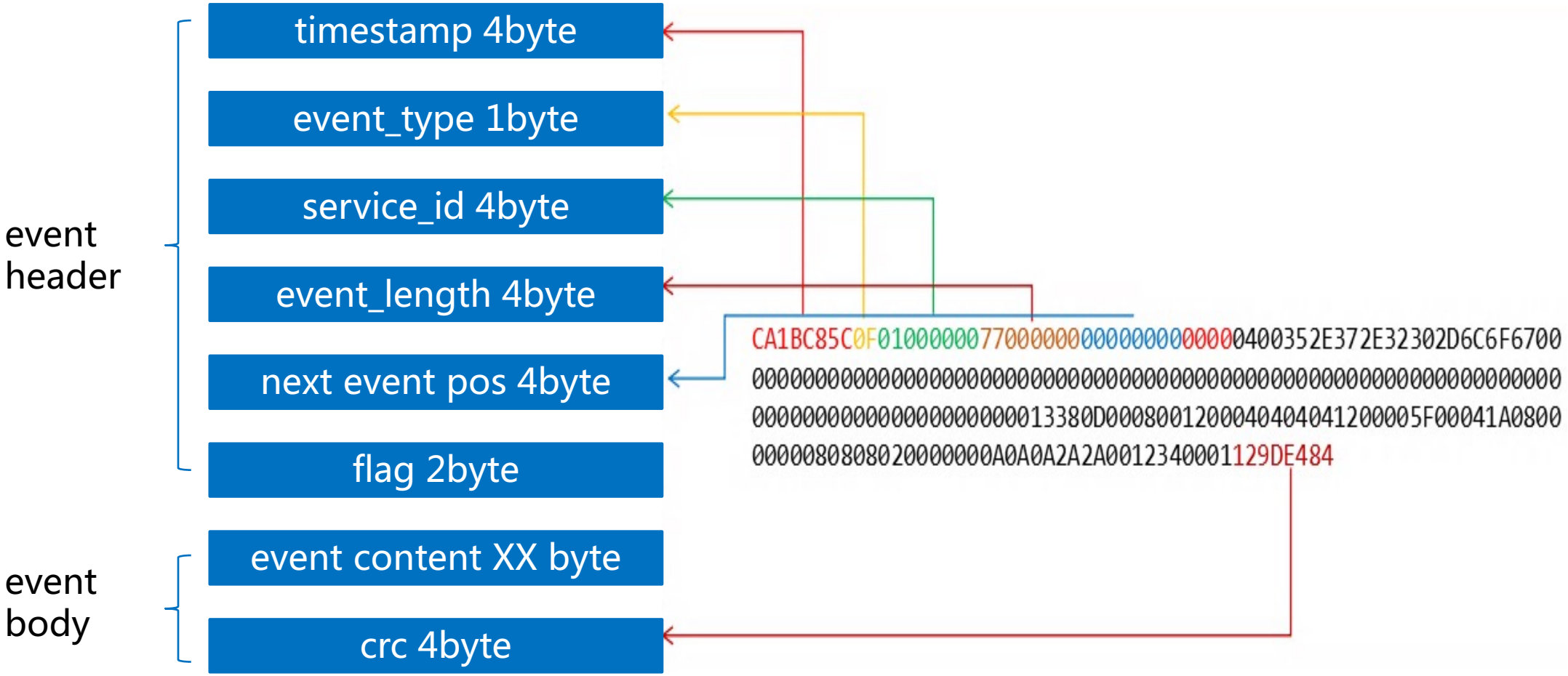
建表语句：

```
create table key.a(
    id int(2) not null,
    name varchar(253) null
);
```

闪回binlog事件解析过程（2/3）

GreatDB 万里数据库

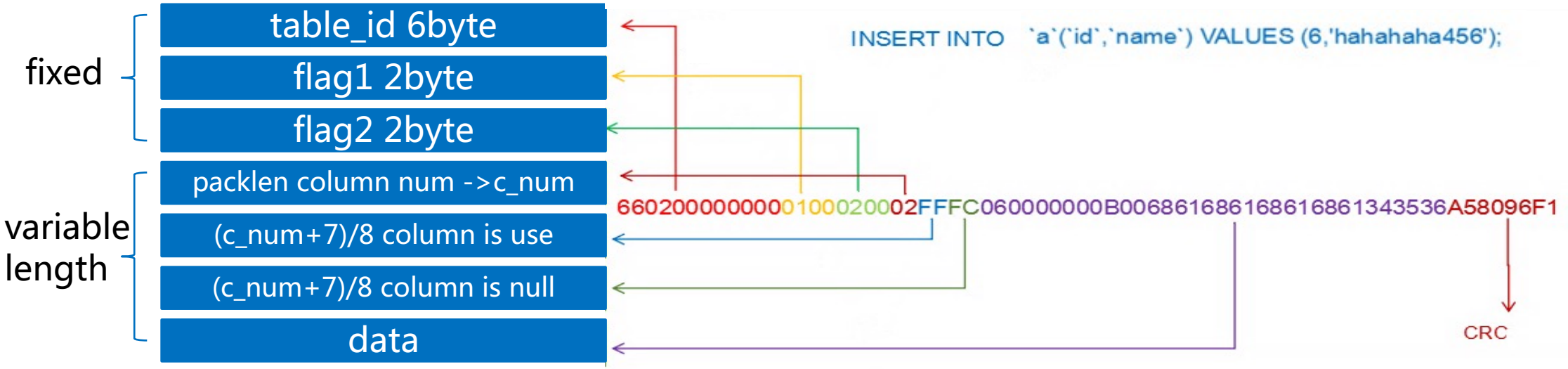
WRITE_ROWS_EVENT事件头结构



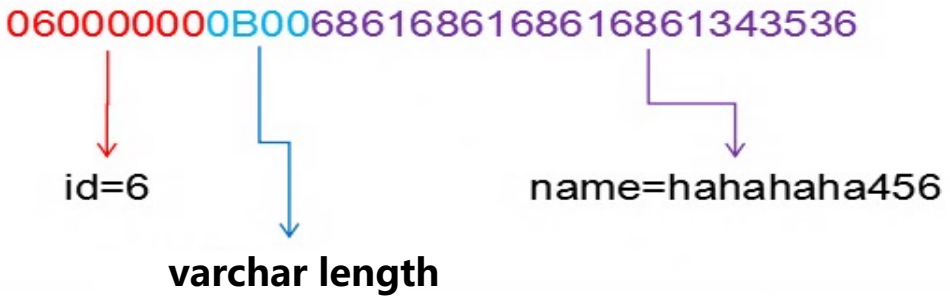
闪回binlog事件解析过程（3/3）

GreatDB 万里数据库

WRITE_ROWS_EVENT事件体结构



数据内容说明：



DML闪回实现步骤

GreatDB 万里数据库

1 数据解析

- 基于row模式的binlog文件，先解析出SQL增删改的操作轨迹。
- 闪回日志格式必须是binlog_format=row，binlog_row_image=full。

2 事件识别

- FORMAT_DESCRIPTION_EVENT
- TABLE_MAP_EVENT
- ROWS_EVENT
- QUERY_EVENT
- XID_EVENT

3 操作反转

- 插入映射成删除
- 删除映射成插入
- 更新交换新旧数据区间

4 顺序反转

- 操作顺序逆序遍历处理

输出

数据回滚binlog文件

GreatDB
万里数据库

目录

CONTENTS

DDL闪回的实现方案

GreatDB 万里数据库

挑战

删表操作是**危险**的行为，而目前主流的MySQL闪回能力主要集中在针对DML语句的回滚还原，对于DDL语句的回滚面临的挑战：

- 文件格式binlog_format=row对于DDL语句的局限性。
- 删表操作在binlog只记录一个sql statement。

思路

- **检测drop table语句转换成先delete再删表**
  - **性能考量：**记录多时，处理性能低。
- **引入表回收站机制**
  - **新建备份#bak_database：**用于保存被删除的历史数据。
  - **监听删表事件：**在删除的动作开始之前，把表数据备份起来，然后留一个空表，在空表上执行“删除”操作。

表回收站实现流程

GreatDB 万里数据库

MySQL删表流程（基于8.0.25）

创建线程接收用户请求

handle_connection（connection_handler_per_thread.cc）

do_command（sql_parse.cc）

dispatch_command（sql_parse.cc）

dispatch_sql_command（sql_parse.cc）

mysql_execute_command（sql_parse.cc）

识别删表sql_command事件：SQLCOM_DROP_TABLE

实现逻辑

- **生成备份表名：**原表+时间戳+uuid/随机数
- **构造备份表标识：**Table_ident(const LEX_CSTRING &db_arg, const LEX_CSTRING &table_arg)
- **备份表加载至Query_block：**add_table_to_list
- **权限校验：**当前用户是否有drop表权限
- **当前表移至备份表：**rename `current_tbl` into `#bak_database`.`current_tbl_xxxxx`

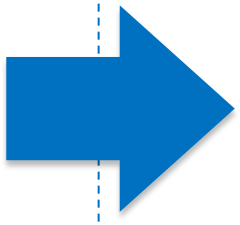
表回收站实现效果

GreatDB 万里数据库

```
mysql> use ccs
Database changed
mysql> select * from test;
+------+------+
| id   | name |
+------+------+
|    1 | nana |
|    1 | nana |
|    1 | nana |
+------+------+
3 rows in set (0.00 sec)

mysql> drop table test;
Query OK, 0 rows affected (0.02 sec)

mysql> select * from test;
ERROR 1146 (42S02): Table 'ccs.test' doesn't exist
mysql>
```



recycle schema

```
mysql> use table_recycle;
Database changed
mysql> show tables;
+----------------------------+
| Tables_in_table_recycle    |
+----------------------------+
| test_20220104162433_44569 |
+----------------------------+
1 row in set (0.01 sec)

mysql> select * from test_20220104162433_44569;
+------+------+
| id   | name |
+------+------+
|    1 | nana |
|    1 | nana |
|    1 | nana |
+------+------+
3 rows in set (0.00 sec)

mysql>
```

recycle schema定义

- **初始化时机：**MySQL数据库启动的时候，初始化一个专用回收站的schema，命名为 "table_recycle"，作为回收站专有database。
- **权限控制：**recycle_bin作为回收站的schema，定义为系统级database，用户没有权限做修改和删除。

谢 谢 聆 听

THANKS FOR YOUR ATTENTION

GreatDB
万里数据库