PGCon 2017

A conference for PostgreSQL
Users and Developers

XLOGMINER
- an Open-Source SQL miner on WAL log

Grant Zhou / grantzhou@highgo.com

@grantchou

May 25, 2017

HighGo Software

www.highgo.com

Background and Goals

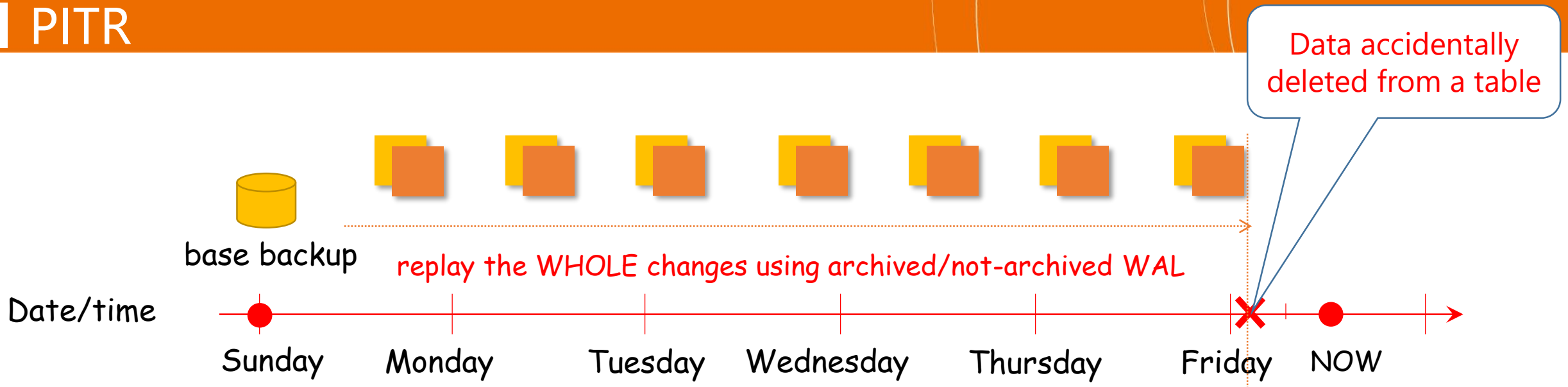
1. At all times, PostgreSQL maintains a write ahead log (**WAL**) in the pg_xlog/ subdirectory of the cluster‘s data directory, it is detail enough for the database recovery.

2. **Logical decoding** is implemented by decoding the contents of the write-ahead log (WAL) into an application-specific form such as a stream of tuples or SQL statements.

Based on these two foundations, it is possible to retrieve the full SQL from WAL, and it will be helpful to the DBA on the followings:

- Partial audit the table for all the data history changes, as long as the corresponding WAL log exists
- Partial recover the changes without PITR
- Similar to “Flashback Query” function in Oracle

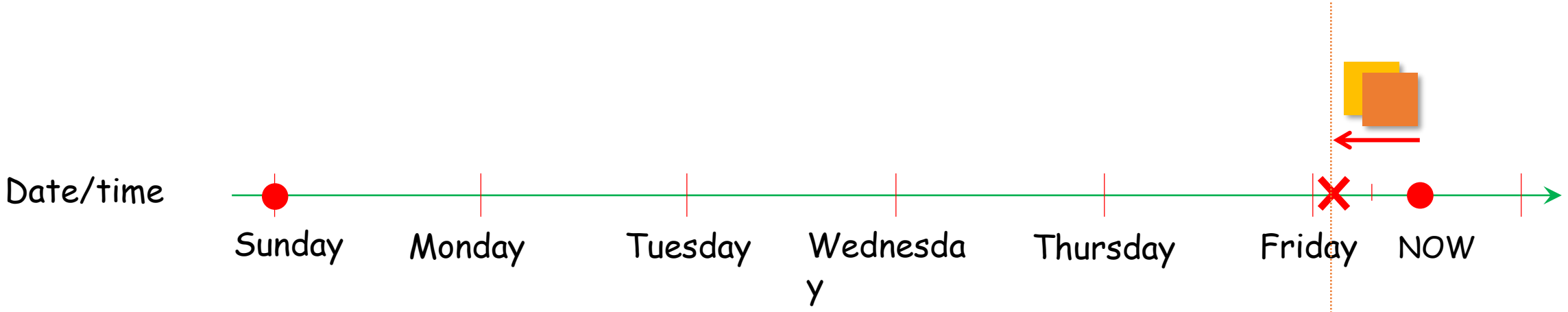
PITR



- Basic backup is necessary
- User have to replay the completely cluster changes, even if only one table is need to be restored. Can not used **for online recovery** or **change checking**.
- It is an **move-forward** process (To recover, you must go back to the previous backup point and then roll forward from there)
- Support ANY recovery with ANY type of operation

https://www.postgresql.org/docs/9.6/static/continuous-archiving.html

XLOGMiner



- Basic backup is NOT necessary, as long as you have enough WAL
- User can parse all the operations on all the tables, and filer the “undo sql” on the specific table only.
- It is an **move-backward** process

XLOGMiner

WAL logs

\+

Current database

XLogMiner

SQL
+
UNDO SQL

xid	1952
virtualxid	1
timestamptz	2017-05-23 14:48:51.895002+08
record_database	highgo
record_user	lchch
record_tablespace	pg_default
record_schema	public
op_type	UPDATE
op_text	**UPDATE "public"."xlogminer_test" SET "att_name" = 'BBB_CHANGE' WHERE "att_id"=2 AND "att_name"='BBBBBBB'**
op_undo	**UPDATE "public"."xlogminer_test" SET "att_name" = 'BBBBBBB' WHERE "att_id"=2 AND "att_name"='BBB_CHANGE' AND ctid = '(0,4)';**

How to use XLogMiner

Setup is the same with logical decoding, for example:
1、 wal_level = logical
2、ALTER TABLE xxx REPLICA IDENTITY FULL;

Then, with extension xlogminer installed, run the SQL as usual

```
// Execute the below test SQL
highgo=# create table xlogminer_test(att_id int, att_name varchar);
CREATE TABLE
highgo=# insert into xlogminer_test values(1,'AAAAAAA');
INSERT 0 1
highgo=# insert into xlogminer_test values(2,'BBBBBBB');
INSERT 0 1
highgo=# insert into xlogminer_test values(3,'CCCCCCC');
INSERT 0 1
highgo=# delete from xlogminer_test where att_id = 1;
DELETE 1
highgo=# update xlogminer_test set att_name = 'BBB_CHANGE' where att_id = 2;
UPDATE 1
highgo=#
```

How to use XLogMiner

Step 2 run the SQL to parse the WAL.
User can specify the exact WAL file, or a directory contains WAL files

```
1  //Step 2. run the parser SQL
2  highgo=# select xlogminer_load_dictionary('/mnt/hgfs/share/test/dic');
3   xlogminer_load_dictionary
4  ---------------------------
5   Dictionary load success!
6  (1 row)
7
8  highgo=# select xlogminer_xlogfile_add('/mnt/hgfs/share/test/list');
9   xlogminer_xlogfile_add
10 ------------------------
11  1 file add success
12 (1 row)
13
14 highgo=# select xlogminer_start('NULL','NULL',0,0);
15  xlogminer_start
16 ------------------
17  xlogminer start!
18 (1 row)
```

Demo / Example

Step 3 Check Result

```
//Step 3. Check the result
highgo=# select xid,op_text,op_undo from xlogminer_contents order by xid,virtualxid;
-[ RECORD 1 ]----------------------------------------------------------------------------------------------------------------------------------
xid       | 1947
op_text   | CREATE TABLE "xlogminer_test"("att_id" int4,"att_name" varchar);
op_undo   | NULL
-[ RECORD 2 ]----------------------------------------------------------------------------------------------------------------------------------
xid       | 1948
op_text   | INSERT INTO "public"."xlogminer_test"("att_id", "att_name") VALUES(1, 'AAAAAAA');
op_undo   | DELETE FROM "public"."xlogminer_test" WHERE "att_id"=1 AND "att_name"='AAAAAAA' AND ctid = '(0,1)';
-[ RECORD 3 ]----------------------------------------------------------------------------------------------------------------------------------
xid       | 1949
op_text   | INSERT INTO "public"."xlogminer_test"("att_id", "att_name") VALUES(2, 'BBBBBBB');
op_undo   | DELETE FROM "public"."xlogminer_test" WHERE "att_id"=2 AND "att_name"='BBBBBBB' AND ctid = '(0,2)';
-[ RECORD 4 ]----------------------------------------------------------------------------------------------------------------------------------
xid       | 1950
op_text   | INSERT INTO "public"."xlogminer_test"("att_id", "att_name") VALUES(3, 'CCCCCCC');
op_undo   | DELETE FROM "public"."xlogminer_test" WHERE "att_id"=3 AND "att_name"='CCCCCCC' AND ctid = '(0,3)';
-[ RECORD 5 ]----------------------------------------------------------------------------------------------------------------------------------
xid       | 1951
op_text   | DELETE FROM "public"."xlogminer_test" WHERE "att_id"=1 AND "att_name"='AAAAAAA'
op_undo   | INSERT INTO "public"."xlogminer_test"("att_id", "att_name") VALUES(1, 'AAAAAAA');
-[ RECORD 6 ]----------------------------------------------------------------------------------------------------------------------------------
xid       | 1952
op_text   | UPDATE "public"."xlogminer_test" SET "att_name" = 'BBB_CHANGE' WHERE "att_id"=2 AND "att_name"='BBBBBBB'
op_undo   | UPDATE "public"."xlogminer_test" SET "att_name" = 'BBBBBBB' WHERE "att_id"=2 AND "att_name"='BBB_CHANGE' AND ctid = '(0,4)'
```

Limits

- PG Kernel need modifications if user need DDL parser function from WAL
- Depends the current database dictionary, modified table structure may cause some parsing failures when XLogMiner parsing the WAL with original table schema, for example:
  - Table dropped - all ins/upd/del can not be parse out
  - User renamed – new user name will be used
  - Column dropped – will be decoded as not readable hex result, like the output of encode ('\x98AE0000', 'hex')
  - ....

Project on Github

- The project is open source, and you can clone the project from Github

- https://github.com/HighgoSoftware/XLogMiner

HighGo Software

CONTACT

@grantchou

grantzhou@highgo.com

2018 156th Ave NE, Bellevue, WA 98007, USA

www.highgo.com