从PostgreSQL实现Flashback功能模块
谈小白如何开始内核定制开发

兰海

武汉大学计算机学院

个人介绍

- 2013年 武汉大学计算机学院 本科 计算机科学与技术 现在大四
- 大二下上彭煜玮老师的《数据库原理》，结识了他
- 大三上学习了《数据库系统实现》同时开始在彭煜玮老师指导下学习《PostgreSQL内核分析》同对PG的部分源码进行了分析也看了些其他PG有关的书。同时，做一些pg的内核的实践。
- 近期在开始研究llvm和GPU上相关内容
- 现在已保研本校本院

目录

1、实现闪回删表功能

2、实现闪回查询功能

3、个人成长和感悟
——如何开始内核定制开发

实现闪回删表

功能需求

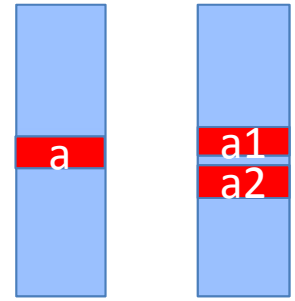
- 在oracle有功能**drop table table_name** 后，能通过sql命令：
  flashback table table_name to before drop ［**rename to new_table_name**］
  将删除的表进行恢复。

- 如果drop table table_name **purge** 则表删除不会通过回收站，即删除后的表是不可以恢复的

实现闪回删表

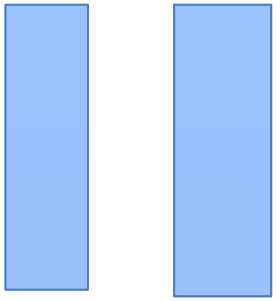
实现思想

- 在PG中要查询一个表上的元组必要的信息包括了该表在pg_class和pg_attribute上对应的元组以及对应的表的数据文件



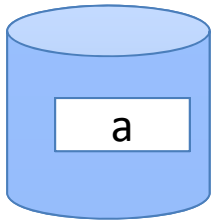
pg_class

pg_attribute

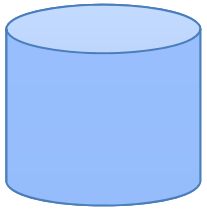


系统表元组回收站

- 要实现闪回删表就需要建立回收站将系统表元组和表文件进行回收，在执行drop table时，就是删除到回收站，如果带purge就是彻底删除
- 元组回收就是将要删除的元组放到回收站中
- 文件回收：将文件移动到回收站中，由于防止oid回卷机制存在，第一个main文件移动后要建立一个空的同名文件作为占位符
- 在Flashback时就是上诉的逆过程



数据文件目录

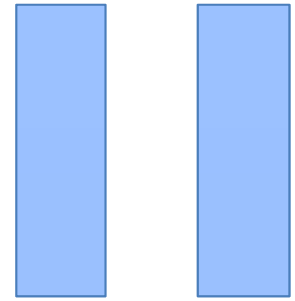


数据文件回收站

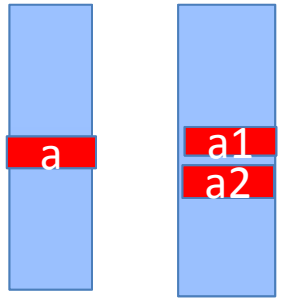
实现闪回删表

实现思想

- 在PG中要查询一个表上的元组必要的信息包括了该表在pg_class和pg_attribute上对应的元组以及对应的表的数据文件



pg_class　　pg_attribute

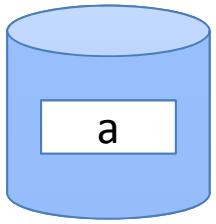


系统表元组回收站

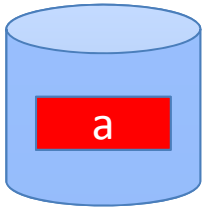
- 要实现闪回删表就需要建立回收站将系统表元组和表文件进行回收，在执行drop table时，就是删除到回收站，如果带purge就是彻底删除

- 元组回收就是将要删除的元组放到回收站中

- 文件回收：将文件移动到回收站中，由于防止oid回卷机制存在，第一个main文件移动后要建立一个空的同名文件作为占位符



数据文件目录



数据文件回收站

- 在Flashback时就是上诉的逆过程

具体实现

- 修改词法、语法使的PG能够识别purge 和 flashback table
- 添加两张新的系统表保存用来保存pg_class和pg_attribute中的对应元组
- 建立回收站目录，同时对initdb作修改，使的初始化时自动建立回收站目录
- 实现drop table功能部分：
  在系统表元组进行删除前，将元组进行拷贝
  在数据文件进行删除时进行移动到回收站，同时新建一个同名的空白文件在原地
- 实现Flashback table功能部分：
  在功能执行分支中加入新的flashback table分支，完成对系统表元组插入pg_class 和pg_attribute，以及完成数据文件的移动

实现闪回删表

效果展示

```
hailan=# select * from test_flatable;
 name | age
------+-----
 张三 |  19
 李四 |  18
 王五 |  21
 江一 |  20
(4 rows)

hailan=# drop table test_flatable;
DROP TABLE
hailan=# select * from test_flatable;
ERROR:  relation "test_flatable" does not exist at character 15
STATEMENT:  select * from test_flatable;
ERROR:  relation "test_flatable" does not exist
LINE 1: select * from test_flatable;
                      ^
```

图-1

```
hailan=# flashback table test_flatable to before drop;
FLASHBACK TABLE
hailan=# select * from test_flatable;
 name | age
------+-----
 张三 |  19
 李四 |  18
 王五 |  21
 江一 |  20
(4 rows)
```

图-2

```
hailan=# drop table test_flatable purge;
DROP TABLE
hailan=# flashback table test_flatable to before drop;
ERROR:  relation "test_flatable" does not exist
STATEMENT:  flashback table test_flatable to before drop;
ERROR:  relation "test_flatable" does not exist
hailan=#
```

图-3

功能需求

- oracle可以在查询时指定查询以前某个时间段的表上的数据，即使表上的数据已经被删掉了，如果满足在在查询时间时存活的

- 通过sql语句：

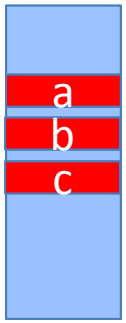
  select * from table_name **as of timestamp**

实现思想

- PG上删除表元组不是真正意义的删除而是进行标记删除。之后在某个时间系统进行autovacuum或者用户进行手动vaccum，实现真正元组删除工作,也就是说旧元组在PG里面仍然存在
- 要记录元组的生命周期，通过生命周期来查看此时的闪回查询的时间点上，该元组是否是活的
- 在扫描表时（闪回查询时），对表元组进行可见性判断时，将可见性判断变成通过时间来判断元组的可见性
- 类似与oracle加入了undo_retention来保证被删元组的最小存活时间

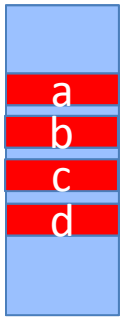
实现闪回查询

实现思想



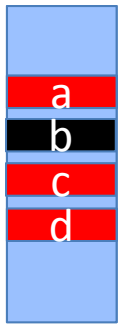
T	B	D
a	t1	-
b	t1	-
c	t1	-

t1



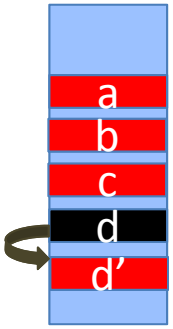
T	B	D
a	t1	-
b	t1	-
c	t1	-
d	t1	-

t2



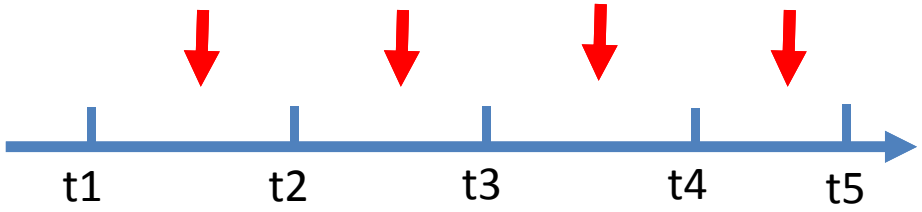
T	B	D
a	t1	-
b	t1	t3
c	t1	-
d	t1	-

t3



T	B	D
a	t1	-
b	t1	t3
c	t1	-
d	t1	t4
d'	t4	-

t4



假设在t5时刻闪回查询

具体实现

- 对词法，语法部分的修改，使得postgresql能够识别asof语法
- 在heaptupdata结构体中加入元组出生和死亡时间存储字段
- 修改元组的插入、更新、删除函数，加入时间量的赋值和更新
- 修改执行计划生成，将带有asof表的扫描计划强制为seqscan，同时将Seqscan中对元组的可见性判断（在asfo情况下）变为对时间点的存活判断来判断元组的可见性
- 设置undo_retention参数，在autovaccum执行代码中利用该参数，在进行对表的标记删除数据进行清理的时候改为有条件的进行清理不是对所有的标记删除的元组都进行清理。需要进行时间检查，如果在时间范围内做fro-zen id 而不删除

实现闪回查询

效果展示

```
hailan=# select * from test_flasera asof timestamp '2016-10-18 13:53:00';
 age
-----
  11
  12
  13
  14
  15
  16
  17
  18
  19
  10
(10 rows)
```

图-1

```
hailan=# select * from test_flasera asof timestamp '2016-10-18 13:55:00';
 age
-----
  11
  12
  13
  14
  15
  16
  17
  18
  19
  10
  99
(11 rows)
```

图-2

```
hailan=# select * from test_flasera asof timestamp '2016-10-18 13:57:00';
 age
-----
  11
  12
  13
  14
  15
  16
  17
  18
  19
  99
 100
(11 rows)
```

图-3

```
hailan=# select * from test_flasera;
 age
-----
  11
  12
  13
  14
  15
  16
  17
  18
  19
  99
 100
(11 rows)
```

图-4

个人成长与感悟

- 学习原理知识
- 实践工作
- 总结自己
- 发现更好玩的事情

个人成长与感悟

理论知识

第一阶段：

《官方文档》 《PostgreSQL修炼之道》

第二阶段：

《数据库系统实现》 《PostgreSQL内核分析》
《数据查询优化器的艺术》

现在：

偶尔还是要翻翻上面的书，看一些理论的论文

实践工作

实践的前提：

1、要有PG内核源码的分析经历

2、要从小的内核修改demo中不断精进

具体的实践：

1、功能分析

2、结合PG内部机制提取出自己的实现方案

3、动手实践

4、调试（编码五分钟，调试两小时）

5、测试（回归测试，压力测试等）

学会总结

一定要有个人技术笔记，思考过程，出现问题的解决方案等等

个人成长与感悟

发现更好玩的事情

llvm/gpu/fpga?

Thanks!

Q & A