

实验使用说明

1. 单一测试

插入操作：剪切板复制，一次性输入，在 `my_table` 中查看最近的两个时间相减即可

```
DROP TABLE IF EXISTS my_table;

CREATE TABLE my_table (
  id SERIAL PRIMARY KEY,
  event_time TIMESTAMP NOT NULL
);

DELETE FROM my_table;

INSERT INTO my_table (event_time) VALUES (CURRENT_TIMESTAMP);
\i /4.sql;

INSERT INTO my_table (event_time) VALUES (CURRENT_TIMESTAMP);

SELECT * FROM my_table;
```

查询/更新/删除操作：选择对应的语句在命令行输入即可

```
EXPLAIN ANALYZE select * from movies_metadata;

EXPLAIN ANALYZE select count(*) from movies_metadata where original_language = 'cn';

EXPLAIN ANALYZE UPDATE movies_metadata SET adult = 'True' WHERE adult = 'False';

EXPLAIN ANALYZE DELETE FROM movies_metadata;
```

2. 多用户测试

双用户：

操作一：一个用户通过运行 `4.sql` 文件进行一段时间连续的文件输入，另一用户不定时利用剪切板快速输入查询指令。

```
--user1

\i /4.sql;

--user2

EXPLAIN ANALYZE select * from movies_metadata;
```

操作二：一个用户通过运行 `4.sql` 文件进行一段时间连续的文件输入，另一用户也运用 `4.sql` 文件进行输入，但统计其插入操作的时间

```
--user1

\i /4.sql;

--user2 同时进行

DROP TABLE IF EXISTS my_table;

CREATE TABLE my_table (
  id SERIAL PRIMARY KEY,
  event_time TIMESTAMP NOT NULL
);

DELETE FROM my_table;
```

```
INSERT INTO my_table (event_time) VALUES (CURRENT_TIMESTAMP);  
  
\i /4.sql;  
  
INSERT INTO my_table (event_time) VALUES (CURRENT_TIMESTAMP);  
  
SELECT * FROM my_table;
```

多用户：用户一运行 4.sql 文件进行一段时间连续的文件输入，用户二和三则不定时的依次利用剪切板快速更新指令

```
\i /4.sql;--user1  
  
EXPLAIN ANALYZE UPDATE movies_metadata SET adult = 'True' WHERE adult = 'False';--user2  
  
EXPLAIN ANALYZE UPDATE movies_metadata SET adult = 'False' WHERE adult = 'True';--user3
```

3. 硬件测试

在对应的数据库中运行 4.sql 文件，进行读入操作，在云监控服务中查看各项硬件的使用情况。

```
\i /4.sql;
```