SUSTech Southern University of Science and Technology

# SQLite数据迁移至openGauss

# 行业背景趋势及解决的痛点

SQLite是一个非常轻量级的数据库， 因此对于手机，相机等设备的嵌入式软件是非常好的选择。但是由于SQLite一般用于处理小到中型数据存储，对于高并发高流量的应用不适用，因此 openGauss是更适合的选择。通过这个数据迁移方案可以高效快速、高稳定性的完成SQLite数据库至openGauss数据库的迁移。

SUSTech Southern University of Science and Technology

# 目录

SUSTech Southern University of Science and Technology

# 团队介绍

团队成员：黄慧惠、牛景萱、邱逸伦、黄梓通
指导老师：朱悦铭

# 方案概述

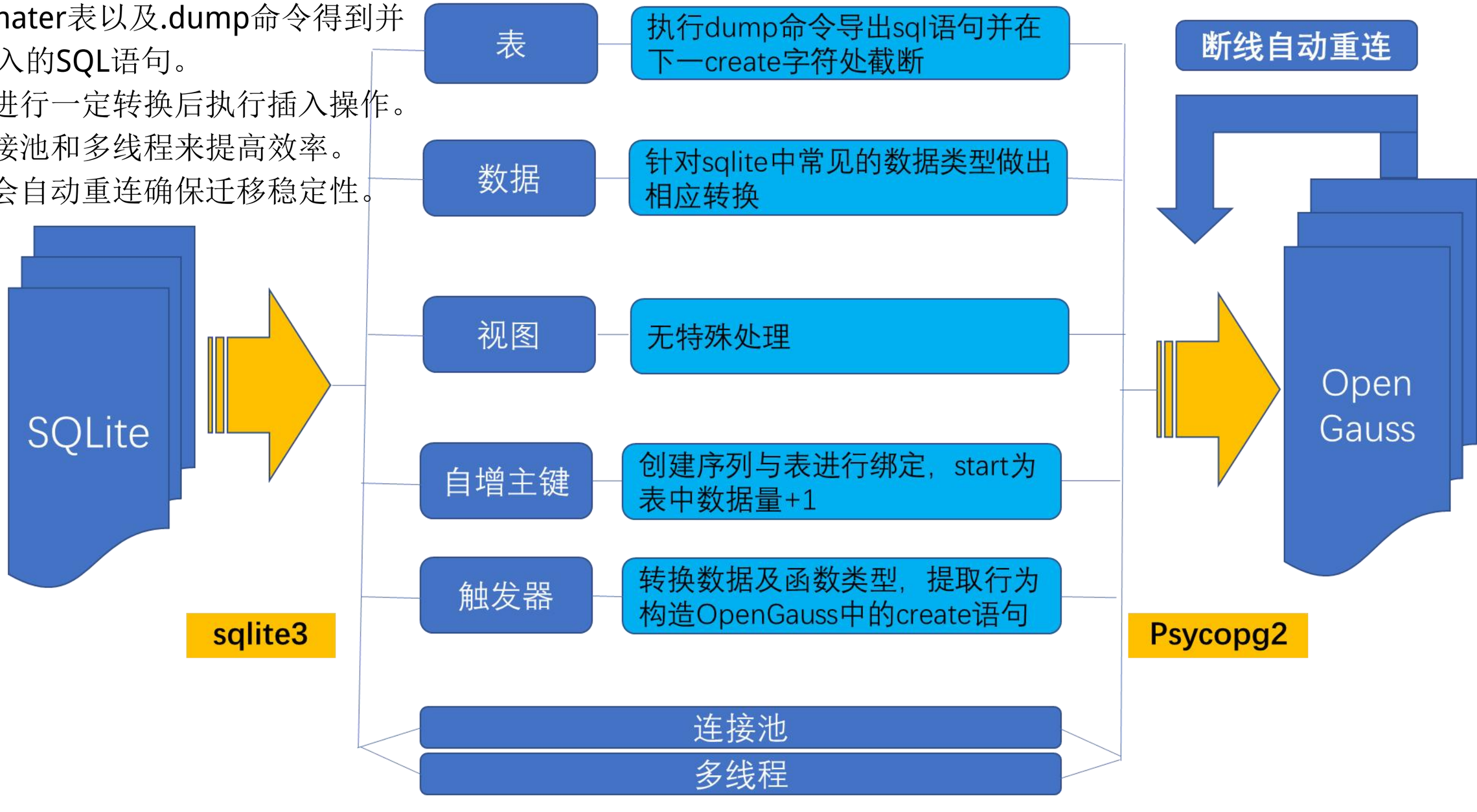
基于 Python 编写的将 SQLite 数据迁移至 openGauss 的数据库迁移工具。支持多种数据类型、索引、视图、触发器、自增序列等，同时使用数据库连接池，实现多线程迁移数据以提高效率。

# 配置介绍

Python版本：Python 3.9.10
openGauss版本：openGauss 2.1.0
SQLite版本：SQLite 3.39.2
服务器系统版本：openEuler 20.03 8vCPUs | 16GiB

# 数据迁移方案介绍

1.结合sqlite_mater表以及.dump命令得到并执行建表和插入的SQL语句。
2.对数据类型进行一定转换后执行插入操作。
3.以及使用连接池和多线程来提高效率。
4.并且断线后会自动重连确保迁移稳定性。



# 类型转换规则

| SQLite | openGauss |
|---|---|
| null | null |
| integer | integer |
| real | double precision |
| text | text |
| blob | blob |
| json | json |
| varchar | varchar |
| char | char |
| nvarchar | nvarchar2 |
| boolean | boolean |
| varying character(n) | character varying(n) |
| datetime | timestamp without time zone |
| graphic(n) | nchar(n) |
| year | integer |
| point | point |
| lesg | lesg |
| box | box |
| path | path |
| polygon | polygon |
| circle | circle |
| array | array |



# 转换规则概览

**表语句** 创建表时去除外键约束，在完成数据迁移后使用ALTER TABLE语句增加外键约束。

**数据** sqlite支持的数据类型较广泛，主要针对常见的数据类型一一对应转换。如datetime转换为timestamp without time zone，real 转换为double precision，nvarchar转换为nvarchar2等，并支持array，json等复杂类型的转换。

**视图** 经大量测试后，已验证对于简单查询和使用sqlite和openGauss共有的函数的视图无需特殊处理可直接导入

**触发器** 通过字符串替换提取sqlite trigger中的行为逻辑，生成openGauss中的function与trigger

**自增主键** 创建相应的序列，并将创建的序列与该字段绑定，start设置为表中数据量+1

**注释** 对带有行内注释--的SQL语句经处理删除了注释进行执行

**索引** 经大量测试后，已验证无需特殊处理可直接导入

# 配置与执行操作介绍

SUSTech Southern University of Science and Technology

```
[root@ecs-ead4 OpenGauss-Project-main]# python main.py
Input the database name of OpenGauss:postgres
Input the schema name of OpenGauss:demo1
Input the host of OpenGauss:120.46.202.38
Input the port of OpenGauss:26000
Input the username of OpenGauss:hhh
Input the user password of OpenGauss:sustechD1
```

## 易于执行

项目基于Python脚本语言构建，客户端主机仅需安装Python3.9环境，运行项目路径下的main.py脚本即可进行SQLite至openGauss数据迁移过程

**执行命令：python main.py**

## 易于配置

通过.properties文件可配置openGauss数据库的名称、主机地址、端口号、存储数据的schema、进行操作的用户等连接信息和SQLite数据库的文件位置

**配置文件名字：opengauss.properties sqlite.properties**

/OpenGauss-Project-main/prop

| 文件名 | 大小 | 类型 |
|---|---|---|
| sqlite.properties | 60 B | Propertie... |
| properties.py | 1.2 KB | Python File |
| opengauss.properties | 139 B | Propertie... |
| _pycache_ | | 文件夹 |

openGauss_opengauss.properties

```
1 database.name=postgres
2 database.schema=demo
3 database.host=120.46.202.38
4 database.port=26000
5 database.user=hhh
6 database.password=sustechD1
7
```

## 个性化命令

执行Python脚本时可以带上多样的参数以选择是否根据.properties文件完成数据迁移、是否以多线程模式进行数据迁移提高迁移效率

**参数命令：**

**-o 选择使用openGauss连接配置文件**

**-s 选择使用SQLite连接配置文件**

**-m 指使用多线程迁移数据**

## 日志库

严格区分的三个日志文件，分别存储了数据库连接的提示信息、迁移时错误信息的提示、以及可选择是否存储的所有被执行的openGauss数据库SQL语句

**日志文件名：**

**info.log error.log sqls.log**

penGauss-Project-main/log

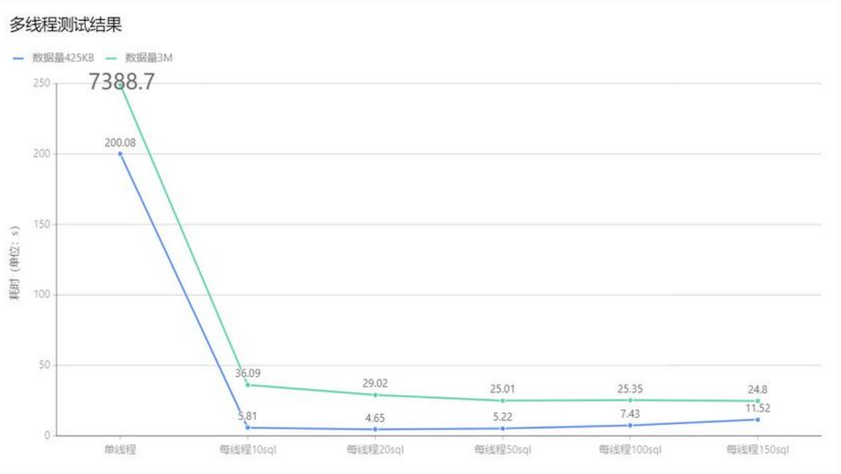
| 文件名 | 大小 |
|---|---|
| sqls.log | 907.9 KB |
| info.log | 882 B |
| error.log | 407 B |

```
[root@ecs-ead4 OpenGauss-Project-main]# python main.py -o opengauss.properties -s sqlite.properties -m
Save the SQL statements in Data Migration? [y/n]y
Successfully Log In OpenGauss Database postgres As hhh
Successfully Log In Sqlite3 Database shenzhen_metro.sqlite
The data migration operation is in progress...
Time Cost = 0.84 seconds
[root@ecs-ead4 OpenGauss-Project-main]#
```

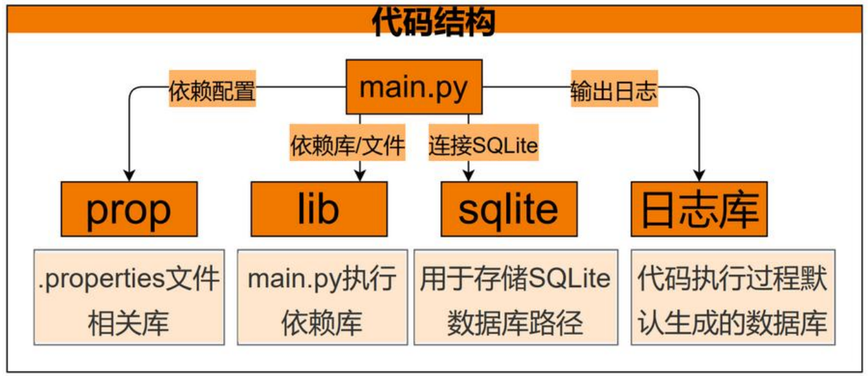
openGauss_sqls.log

```
39 [2022-08-29 11:46:53] [INFO] >>> INSERT INTO "bus_lines" VALUES(1,'M373');
40 [2022-08-29 11:46:53] [INFO] >>> INSERT INTO "bus_lines" VALUES(1,'M401');
41 [2022-08-29 11:46:53] [INFO] >>> INSERT INTO "bus_lines" VALUES(1,'M468');
42 [2022-08-29 11:46:53] [INFO] >>> INSERT INTO "bus_lines" VALUES(1,'N2');
43 [2022-08-29 11:46:53] [INFO] >>> INSERT INTO "bus_lines" VALUES(1,'N4');
44 [2022-08-29 11:46:53] [INFO] >>> INSERT INTO "bus_lines" VALUES(1,'N7');
45 [2022-08-29 11:46:53] [INFO] >>> INSERT INTO "bus_lines" VALUES(1,'N14');
46 [2022-08-29 11:46:53] [INFO] >>> INSERT INTO "bus_lines" VALUES(1,'N15');
```

# 技术亮点以及优势

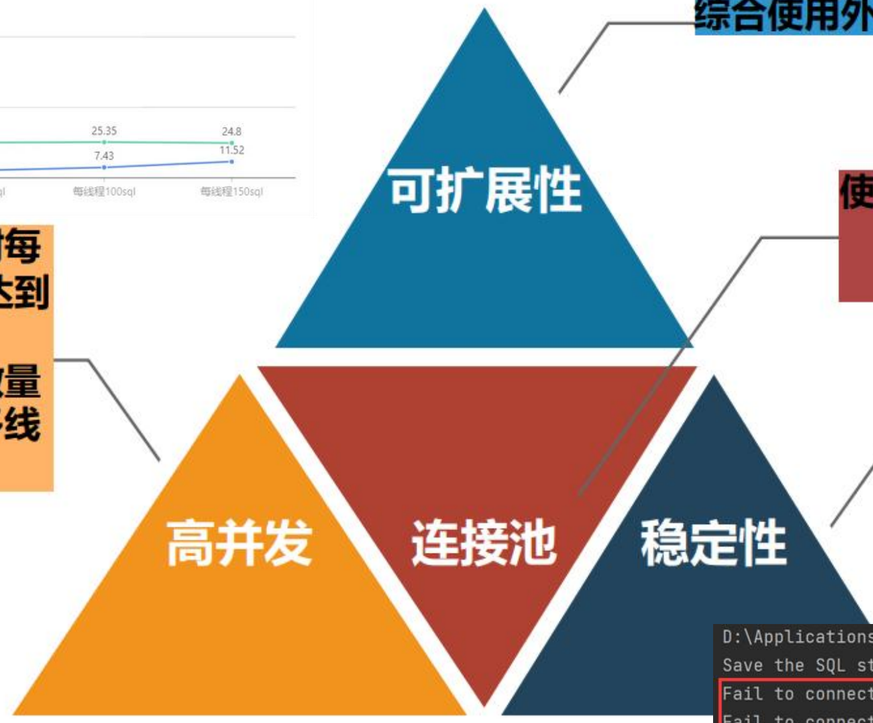



①经多次测试当数据规模过大时每50个sql语句创建一个线程可以达到最优的迁移效率。
②可以自行根据数据库的迁移数量以及电脑配置来设置是否使用多线程高并发操作




设计模式
综合使用外观模式与装饰器模式

```
self.pool = OpenGaussConnectionPool(1, 300,
                                    database=self.opengauss_properties['database.name'],
                                    user=self.opengauss_properties['database.user'],
                                    password=self.opengauss_properties[
                                        'database.password'],
                                    host=self.opengauss_properties['database.host'],
                                    port=self.opengauss_properties['database.port'],
                                    keepalives=1,
                                    keepalives_idle=30,
                                    keepalives_interval=10,
                                    keepalives_count=15)
```




使用psycopg2包实现数据库连接池
最小连接数量为1
最大连接数量为300

①在连接OpenGauss数据库的用户权限足够时，能保证适配各种类型的数据进行迁移
②基于数据库连接池管理连接能保证连接的稳定性并及时释放无用连接给等待的线程
③初始化数据库连接时失败时会进行多次尝试，迁移过程网络连接出错会尝试重新连接

```
D:\Applications\Python\python.exe "D:/1A/Python lab/OpenGauss-Project/main.py" -o opengauss.properties -s sqlite.properties -m
Save the SQL statements in Data Migration? [y/n]y
Fail to connect to OpenGauss database. Retry 1 time
Fail to connect to OpenGauss database. Retry 2 times
Fail to connect to OpenGauss database. Retry 3 times
Successfully Log In OpenGauss Database db_olin As olin
Successfully Log In Sqlite3 Database shenzhen_metro.sqlite
The data migration operation is in progress...
Time Cost = 8.83 seconds

Process finished with exit code 0
```

SUSTech Southern University of Science and Technology