

mysql2opengauss

本文档是 mysql2opengauss 的使用说明文档，通过本文档介绍，能够帮助相关人员成功地把数据从 MySQL 复制到 openGauss。

目 录

第 1 章	工具介绍.....	5
第 2 章	安装方法.....	6
2.1	介质安装.....	6
2.2	源码安装.....	6
第 3 章	配置文件说明	7
3.1	mysql2opengauss. prop.....	7
3.2	metrics.prop	8
3.3	createtablerkv. prop	9
3.4	createfunctionrkv. prop	10
3.5	createfunctiontkv. prop	10
3.6	odbc. ini. template.....	11
3.7	odbcinst. ini. template.....	11
第 4 章	类型转换.....	12
4.1	类型转换规则	12
4.2	自增主键转换	13
4.3	存储过程转换	13
4.4	函数转换.....	13
第 5 章	其他说明.....	14
5.1	执行 sql 出现网络错误时会重试	14
5.2	非空字段约束的 mysql 的空字符串'' 数值会被替换为单一空格的字符串'' ...	14
第 6 章	实例.....	15
6.1	解压介质.....	15
6.2	日志库准备	15
6.3	修改 mysql2opengauss. prop 的连接配置和数据库	15

6.4	在 bin 目录执行 ./run.sh 进行数据迁移.....	16
6.5	grafana 配置.....	17
6.6	通过 grafana 查看日志库信息	18

第1章 工具介绍

本工具是一个用 c/c++编写的 MySQL 到 openGauss 的数据库复制工具。工具基于 mysqldump 编写，通过 odbc 往 openGauss 复制全量数据。支持字段类型、主键、索引、数据类型的自动转换，并且支持视图、存储过程、函数的迁移。在迁移的同时，可以把信息存放到日志库，方便使用 grafana 之类的工具进行查看。

第2章 安装方法

2.1 介质安装

目前提供了 x86 的 centos7 的二进制介质以及 arm 的 openeuler 的二进制介质

在 x86 平台下，解压 `mysqltoopengauss.centos7.x86_64.tar.gz` 即可

在 arm 平台下，解压 `mysqltoopengauss.openeuler.aarch64.tar.gz` 即可

2.2 源码安装

在 x86 平台执行 `build_x86.sh` 进行编译，编译完毕后，在 `output` 下生成安装介质

在 arm 平台执行 `build_aarch64.sh` 进行编译，编译完毕后，在 `output` 下生成安装介质

（注需要 `cmake` 和 `gcc` 编译环境）

第3章 配置文件说明

3.1 mysql2opengauss. prop

这是基础的配置文件，里面配置了 mysql 和 opengauss 的用户名密码以及要迁移数据的源数据库和目的数据库。

配置文件示例如下：

```
mysql_host=127.0.0.1
mysql_port=3306
mysql_username=username
mysql_password=password
mysql_database=database
opengauss_host=127.0.0.1
opengauss_port=5432
opengauss_username=opengauss
opengauss_password=password
opengauss_database=database
loglevel=4
```

每一项配置的含义如下：

mysql_host	mysql 的主机地址
mysql_port	mysql 的端口
mysql_username	mysql 的用户名
mysql_password	mysql 的密码
mysql_database	要迁移的 mysql 源数据库
opengauss_host	openGauss 的主机地址
opengauss_port	openGauss 的端口
opengauss_username	openGauss 的用户名
opengauss_password	openGauss 的密码
opengauss_database	openGauss 的目的数据库
loglevel	打印级别，3 以下（包含 3）会打印执行的 sql 语句

3.2 metrics.prop

这是日志库的配置文件，里面配置了日志库的用户名密码以及日志库类型，目前只日志库支持 opengauss 和 mysql。

配置文件示例如下：

```
metrics_host=127.0.0.1
metrics_port=3306
metrics_username=root
metrics_password=password
metrics_database=metricsdb
metrics_dbtype=mysql
```

每一项配置的含义如下：

metrics_host	日志库的主机地址
metrics_port	日志库的端口
metrics_username	日志库的用户名
metrics_password	日志库的密码
metrics_dbtype	日志库类型，目前只支持 mysql 或 opengauss

注：日志库需要执行 sql 目录下的 sql 创建日志表

3.3 createtablerkv.prop

创建表语句的基础替换文件，是 key-value 的格式，一般不需要修改

配置文件的示例如下：

```
CHARACTER SET utf8 COLLATE utf8_bin=
CHARACTER SET utf8=
COLLATE utf8_bin=
CHARACTER SET latin1 COLLATE latin1_bin=
CHARACTER SET latin1=
COLLATE latin1_bin=
" longtext=" text
tinyint(=integer(
bigint(=integer(
datetime(3)=timestamp without time zone
" datetime=" timestamp without time zone
" timestamp,=" timestamp without time zone,
" timestamp NULL DEFAULT NULL,=" timestamp without time zone NULL,
" timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,=" timestamp without time zone NOT NULL,
" timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',=" timestamp without time zone NOT NULL,
" tinytext=" text
" longblob=" bytea
" mediumtext=" text
mediumint(=integer(
smallint(=integer(
" time,=" time without time zone,
" time NOT NULL,=" time without time zone NOT NULL,
" time DEFAULT NULL,=" time without time zone DEFAULT NULL,
" tinyblob=" bytea
" mediumblob=" bytea
varbinary(=bytea(
binary(=bytea(
decimal(=numeric(
" double,=" double precision,
" double DEFAULT=" double precision DEFAULT
" double NOT NULL=" double precision NOT NULL
" float=" double precision
bit(=integer(
year(=integer(
" geometrycollection=" bytea
" geometry=" point
" linestring=" path
" multipoint=" bytea
" multilinestring=" bytea
" multipolygon=" bytea
int(=integer(
```

3.4 createfunctionrkv.prop

创建函数语句的基础替换文件，是 key-value 的格式，一般不需要修改配置文件的示例如下：

```
) CHARSET latin1_bin=)
) CHARSET latin1=)
) CHARSET utf8_bin=)
) CHARSET utf8=)
```

3.5 createfunctiontkv.prop

创建函数语句的类型替换文件，是 key-value 的格式，一般不需要修改配置文件的示例如下：

```
RETURNS longtext=RETURNS text
RETURNS tinyint(=RETURNS integer(
RETURNS bigint(=RETURNS integer(
RETURNS datetime(3)=RETURNS timestamp
RETURNS datetime=RETURNS timestamp
RETURNS tinytext=RETURNS text
RETURNS longblob=RETURNS bytea
RETURNS mediumtext=RETURNS text
RETURNS mediumint(=RETURNS integer(
RETURNS smallint(=RETURNS integer(
RETURNS tinyblob=RETURNS bytea
RETURNS mediumblob=RETURNS bytea
RETURNS varbinary(=RETURNS bytea(
RETURNS binary(=RETURNS bytea(
RETURNS decimal(=RETURNS numeric(
RETURNS double=RETURNS double precision
RETURNS float=RETURNS double precision
RETURNS bit(=RETURNS integer(
RETURNS year(=RETURNS integer(
RETURNS geometrycollection=RETURNS bytea
RETURNS geometry=RETURNS point
RETURNS linestring=RETURNS path
RETURNS multipoint=RETURNS bytea
RETURNS multilinestring=RETURNS bytea
RETURNS multipolygon=RETURNS bytea
RETURNS int(=RETURNS integer(
```

3.6 odbc.ini.template

odbc 连接模板文件，一般不需要修改

配置文件的示例如下：

```
[opengauss]  
Driver=opengauss  
Servername=SERVERNAME  
Database=DATABASE  
Username=USERNAME  
Password=PASSWORD  
Port=PORT  
Sslmode=disable
```

3.7 odbcinst.ini.template

odbc 驱动模板文件，一般不需要修改

配置文件的示例如下：

```
[opengauss]  
Driver64=BASEDIR/lib/psqlodbcw.so  
setup=BASEDIR/lib/psqlodbcw.so
```

第4章 类型转换

4.1 类型转换规则

转换过来的 bytea 都没有设置长度限制

set (set1, set2, ...) 会直接转换为 text

MySQL	openGauss
tinyint	integer
bigint	integer
mediumint	integer
smallint	integer
int	integer
bit	integer
datetime	timestamp without time zone
datetime(3)	timestamp without time zone
timestamp	timestamp without time zone
time	time without time zone
tinytext	text
longtext	text
longblob	bytea
mediumtext	text
tinyblob	bytea
mediumblob	bytea
varbinary	bytea
binary	bytea
decimal	numeric
double	double precision
float	double precision
year	integer
geometrycollection	bytea
geometry	point
linestring	path
multipoint	bytea
multilinestring	bytea
multipolygon	bytea
set	text

4.2 自增主键转换

如果表中存在自增主键，那么将会增加一个名为 `seq_表名_字段名` 的序列，然后建表时会把这个序列指定到表的自增主键字段，并且会关联这个字段。

4.3 存储过程转换

mysql 的存储过程到 openGauss 的存储过程的转换主要是在定义的 `begin` 之前加上 `IS` 关键字。

4.4 函数转换

mysql 的存储过程到 openGauss 的存储过程的转换主要做了以下两步

- a) 使用 `AS $$` 和 `;$ LANGUAGE plpgsql` 把函数体包起来
- b) 替换配置文件 `createfunctionrkv.prop` 中的定义格式
- c) 替换配置文件 `createfunctiontkv.prop` 中的类型的返回格式

第5章 其他说明

5.1 执行 sql 出现网络错误时会重试

往 openGauss 执行 sql 碰到网络错误 *"could not send data"* 或者 *"no connection"* 时，会尝试重新连接 openGauss，并重新执行这个 sql。最多会重试 30 次，每次间隔 2 秒中。

5.2 非空字段约束的 mysql 的空字符串'' 数值会被替换为单一空格的字符串' '

如果 mysql 中定义了 NOT NULL 的 varchar 字段，并且字段对应的内容为空字符串''，那么由于 openGauss 的 NOT NULL 的 varchar 字段无法插入空字符串''，因此会把空字符串'' 替换为单一空格的字符串' '再进行插入。

第6章 实例

6.1 解压介质

```
tar -xf mysqltoopengauss.centos7.x86_64.tar.gz
```

```
[bes@OPS-ARCHITECT-2-30 bes]$ tar -xf mysqltoopengauss.centos7.x86_64.tar.gz
[bes@OPS-ARCHITECT-2-30 bes]$ ls
mysqltoopengauss  mysqltoopengauss.centos7.x86_64.tar.gz
[bes@OPS-ARCHITECT-2-30 bes]$ cd mysqltoopengauss/
[bes@OPS-ARCHITECT-2-30 mysqltoopengauss]$ ls
bin  conf  lib
[bes@OPS-ARCHITECT-2-30 mysqltoopengauss]$
```

6.2 日志库准备

请把 sql 目录下的在日志库中执行，如果日志库使用 opengauss 请执行 metrics_opengauss.sql，使用 mysql 请执行 metrics_mysql.sql

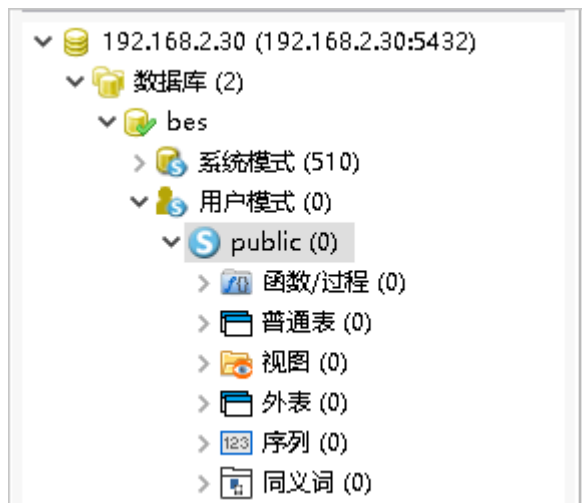
6.3 修改 mysql2opengauss.prop 的连接配置和数据库

比如从 MySQL 的 amdb 数据库迁移数据到 openGauss 的 bes 数据库中，配置修改如下

```
mysql_host=192.168.2.30
mysql_port=3306
mysql_username=username
mysql_password=password
mysql_database=amdb
opengauss_host=192.168.2.30
opengauss_port=5432
opengauss_username=bes
opengauss_password=password
opengauss_database=bes
loglevel=4
```

6.4 在 bin 目录执行 ./run.sh 进行数据迁移

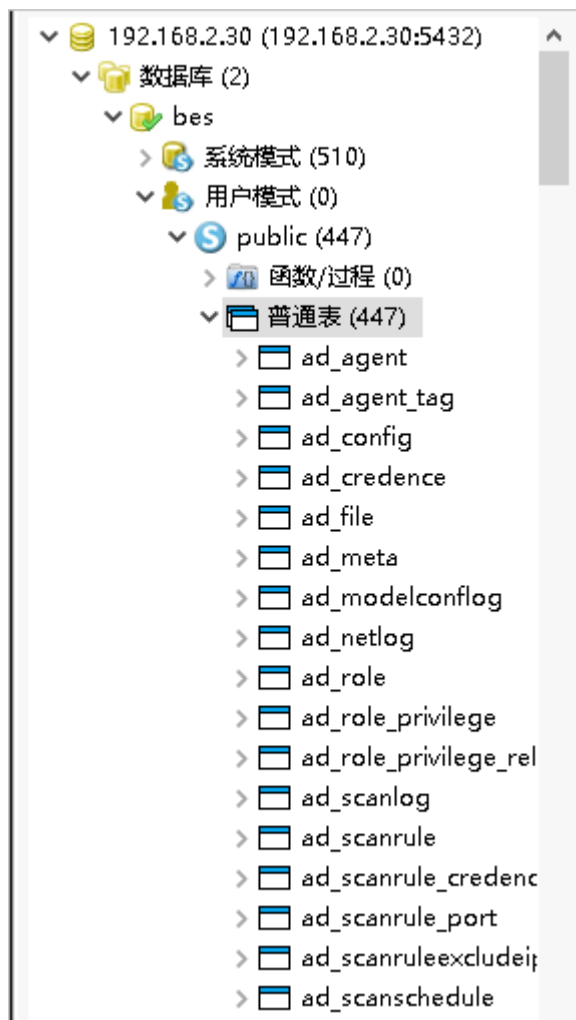
迁移前的 openGauss 数据库如下：



在 bin 目录执行 ./run.sh 进行数据迁移

```
[bes@OPS-ARCHITECT-2-30 bin]$  
[bes@OPS-ARCHITECT-2-30 bin]$ pwd  
/new/bes/mysqltoopengauss/bin  
[bes@OPS-ARCHITECT-2-30 bin]$ ./run.sh  
success.  
[bes@OPS-ARCHITECT-2-30 bin]$
```

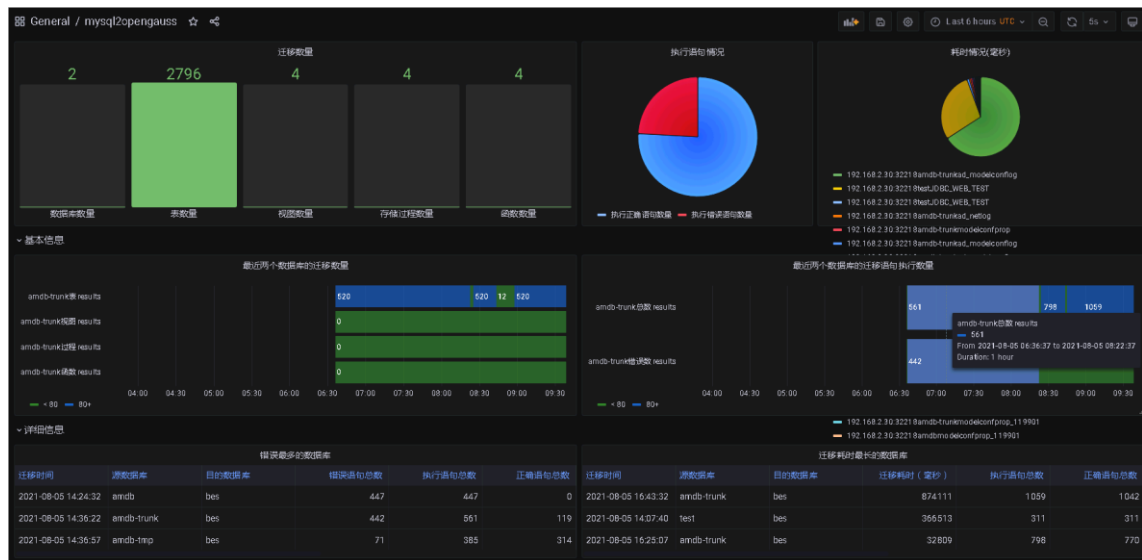

迁移后的 openGauss 数据库如下：



6.5 grafana 配置

可参考 grafana 目录下的 samplemodel.json 进行 grafana 配置

6.6 通过 grafana 查看日志库信息



错误最多的数据库						迁移耗时最长的数据库					
迁移时间	源数据库	目的数据库	错误语句总数	执行语句总数	正确语句总数	迁移时间	源数据库	目的数据库	迁移耗时 (毫秒)	执行语句总数	正确语句总数
2021-08-05 14:24:32	amdb	bes	447	447	0	2021-08-05 16:43:32	amdb-trunk	bes	874111	1059	1042
2021-08-05 14:35:22	amdb-trunk	bes	442	561	119	2021-08-05 14:07:40	test	bes	366513	311	311
2021-08-05 14:35:57	amdb-trmp	bes	71	385	314	2021-08-05 16:25:07	amdb-trunk	bes	32809	798	770

错误最多的表						迁移耗时最长的表					
迁移时间	表名	错误语句总数	执行语句总数	正确语句总数	源数据库	迁移时间	表名	迁移耗时 (毫秒)	源数据库	目的数据库	执行语句总数
2021-08-05 14:23:54	testtx	2	2	0	test1	2021-08-05 16:43:32	ad_netlog	7424	amdb-trunk	bes	7
2021-08-05 14:58:45	testtx	2	2	0	test1	2021-08-05 16:25:07	modelconfprop	7155	amdb-trunk	bes	2
2021-08-05 15:19:34	testtx	2	2	0	test	2021-08-05 16:25:07	ad_modelconflog	4398	amdb-trunk	bes	1

最近的sql错误											
迁移时间	名称	类型	错误信息	sql语句	源数据库地址	目的数据库地址	源数据库	目的数据库	开始时间	结束时间	执行耗时 (毫秒)
2021-08-05 16:43:32	ad_task_tool	table	42F01:ERROR:rela...	CREATE TABLE "ad_...	192.168.2.30:32218	192.168.2.30:5432	amdb-trunk	bes	2021-08-05 16:58:04	2021-08-05 16:58:04	37
2021-08-05 16:43:32	modelpropdef	table	42F01:ERROR:rela...	CREATE TABLE "mo...	192.168.2.30:32218	192.168.2.30:5432	amdb-trunk	bes	2021-08-05 16:58:03	2021-08-05 16:58:03	22
2021-08-05 16:43:32	modelrelation	table	42F01:ERROR:rela...	CREATE TABLE "mo...	192.168.2.30:32218	192.168.2.30:5432	amdb-trunk	bes	2021-08-05 16:58:03	2021-08-05 16:58:03	22
2021-08-05 16:43:32	tag	table	42F01:ERROR:rela...	CREATE TABLE "tag...	192.168.2.30:32218	192.168.2.30:5432	amdb-trunk	bes	2021-08-05 16:58:03	2021-08-05 16:58:03	20
2021-08-05 16:43:32	modelconfrrelation	table	42F01:ERROR:rela...	CREATE TABLE "mo...	192.168.2.30:32218	192.168.2.30:5432	amdb-trunk	bes	2021-08-05 16:58:00	2021-08-05 16:58:00	34
2021-08-05 16:43:32	modelconfprop	table	42F01:ERROR:rela...	CREATE TABLE "mo...	192.168.2.30:32218	192.168.2.30:5432	amdb-trunk	bes	2021-08-05 16:57:55	2021-08-05 16:57:55	33
2021-08-05 16:43:32	model	table	42F01:ERROR:rela...	CREATE TABLE "mo...	192.168.2.30:32218	192.168.2.30:5432	amdb-trunk	bes	2021-08-05 16:57:51	2021-08-05 16:57:51	29
2021-08-05 16:43:32	modelconf	table	42F01:ERROR:rela...	CREATE TABLE "mo...	192.168.2.30:32218	192.168.2.30:5432	amdb-trunk	bes	2021-08-05 16:57:51	2021-08-05 16:57:51	33
2021-08-05 16:43:32	ad_script	table	42F01:ERROR:rela...	CREATE TABLE "ad_...	192.168.2.30:32218	192.168.2.30:5432	amdb-trunk	bes	2021-08-05 16:57:50	2021-08-05 16:57:50	30
2021-08-05 16:43:32	ad_task_tool	table	42F01:ERROR:rela...	CREATE TABLE "ad_...	192.168.2.30:32218	192.168.2.30:5432	amdb-trunk	bes	2021-08-05 16:57:50	2021-08-05 16:57:50	34