**基于 OGG 的 Oracle 与 Hadoop 集群准实时同步介绍**

[[](https://www.qcloud.com/community/user/377958)王亮](https://www.qcloud.com/community/user/377958) 2016-11-15 1399 标签： [Oracle](https://www.qcloud.com/community/tag/150) ， [Hadoop](https://www.qcloud.com/community/tag/144) ， [腾讯云数据库团队](https://www.qcloud.com/community/tag/134)

Oracle 里存储的结构化数据导出到 Hadoop 体系做离线计算是一种常见数据处置手段。近期有场景需要做 Oracle 到 Hadoop 体系的实时导入，这里以此案例做以介绍。

Oracle 作为商业化的数据库解决方案，自发性的获取数据库事务日志等比较困难，故选择官方提供的同步工具 OGG （ Oracle GoldenGate ）来解决。

**安装与基本配置**

**环境说明**

软件配置

| **角色** | **数据存储服务及版本** | **OGG版本** | **IP** |
| --- | --- | --- | --- |
| 源服务器 | OracleRelease11.2.0.1 | Oracle GoldenGate 11.2.1.0 for Oracle on Linux x86-64 | 10.0.0.25 |
| 目标服务器 | Hadoop 2.7.2 | Oracle GoldenGate for Big Data 12.2.0.1 on Linux x86-64 | 10.0.0.2 |

以上源服务器上OGG安装在Oracle用户下，目标服务器上OGG安装在root用户下。

**注意**

Oracle导出到异构的存储系统，如MySQL，DB2，PG等以及对应的不同平台，如AIX，Windows，Linux等官方都有提供对应的Oracle GoldenGate版本，可在[这里](http://www.oracle.com/technetwork/middleware/goldengate/downloads/index.html" \t "_blank)或者在[旧版本](http://edelivery.oracle.com/)查询下载安装。

**Oracle 源端基础配置**

将下载到的对应OGG版本放在方便的位置并解压，本示例Oracle源端最终的解压目录为/u01/gg。

1. 配置环境变量
2. 这里的环境变量主要是对执行OGG的用户添加OGG相关的环境变量，本示例为Oracle用户添加的环境变量如下：（/home/oracle/.bash\_profile文件）
3. export OGG\_HOME=/u01/gg/
4. export LD\_LIBRARY\_PATH=$ORACLE\_HOME/lib:$OGG\_HOME:/lib:/usr/lib
5. export CLASSPATH=$ORACLE\_HOME/jdk/jre:$ORACLE\_HOME/jlib:$ORACLE\_HOME/rdbms/jlib
6. Oracle打开归档模式  
   使用如下命令查看当前是否为归档模式（archive）
7. SQL> archive log list
8. Database log mode Archive Mode
9. Automatic archival Enabled
10. Archive destination /u01/arch\_log
11. Oldest online log sequence 6
12. Next log sequence to archive 8
13. Current log sequence 8

如非以上状态，手动调整即可

SQL> conn / as sysdba(以DBA身份连接数据库)

SQL> shutdown immediate(立即关闭数据库)

SQL> startup mount(启动实例并加载数据库，但不打开)

SQL> alter database archivelog(更改数据库为归档模式)

SQL> alter database open(打开数据库)

SQL> alter system archive log start(启用自动归档)

1. Oracle打开日志相关  
   OGG基于辅助日志等进行实时传输，故需要打开相关日志确保可获取事务内容。通过一下命令查看当前状态：
2. SQL> select force\_logging, supplemental\_log\_data\_min from v$database;
3. FOR SUPPLEME--- --------
4. YES YES

如果以上查询结果非YES，可通过以下命令修改状态：

SQL> alter database force logging;

SQL> alter database add supplemental log data;

1. Oracle创建复制用户  
   为了使Oracle里用户的复制权限更加单纯，故专门创建复制用户，并赋予dba权限
2. SQL> create tablespaceoggtbsdatafile '/u01/app/oracle/oradata/orcl/oggtbs01.dbf' size 1000M autoextend on;
3. SQL> create user ggs identified by ggs default tablespaceoggtbs;
4. User created.
5. SQL> grant dba to ggs;
6. Grant succeeded.

最终这个ggs帐号的权限如下所示：

SQL> select \* from dba\_sys\_privs where GRANTEE='GGS';

GRANTEE PRIVILEGE ADM

GGS DROP ANY DIRECTORY NO

GGS ALTER ANY TABLE NO

GGS ALTER SESSION NO

GGS SELECT ANY DICTIONARY NO

GGS CREATE ANY DIRECTORY NO

GGS RESTRICTED SESSION NO

GGS FLASHBACK ANY TABLE NO

GGS UPDATE ANY TABLE NO

GGS DELETE ANY TABLE NO

GGS CREATE TABLE NO

GGS INSERT ANY TABLE NO

GRANTEE PRIVILEGE ADM

GGS UNLIMITED TABLESPACE NO

GGS CREATE SESSION NO

GGS SELECT ANY TABLE NO

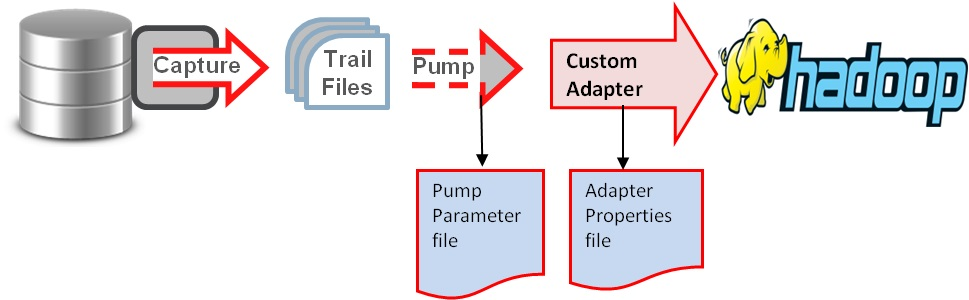
1. OGG初始化  
   进入OGG的主目录执行./ggsci，进入OGG命令行
2. [oracle@VM\_0\_25\_centos gg]$ ./ggsci
3. Oracle GoldenGate Command Interpreter for Oracle
4. Version 11.2.1.0.3 14400833 OGGCORE\_11.2.1.0.3\_PLATFORMS\_120823.1258\_FBO
5. Linux, x64, 64bit (optimized), Oracle 11g on Aug 23 2012 20:20:21
6. Copyright (C) 1995, 2012, Oracle and/or its affiliates. All rights reserved.
7. GGSCI (VM\_0\_25\_centos) 1>
8. 执行create subdirs进行目录创建
9. GGSCI (VM\_0\_25\_centos) 4> create subdirs
10. Creating subdirectories under current directory /u01/gg
11. Parameter files /u01/gg/dirprm: already exists
12. Report files /u01/gg/dirrpt: already exists
13. Checkpoint files /u01/gg/dirchk: already exists
14. Process status files /u01/gg/dirpcs: already exists
15. SQL script files /u01/gg/dirsql: already exists
16. Database definitions files /u01/gg/dirdef: already exists
17. Extract data files /u01/gg/dirdat: already exists
18. Temporary files /u01/gg/dirtmp: already exists
19. Stdout files /u01/gg/dirout: already exists
20. Oracle创建模拟复制库表  
    模拟建一个用户叫tcloud，密码tcloud，同时基于这个用户建一张表，叫t\_ogg。
21. SQL> create user tcloud identified by tcloud default tablespace users;
22. User created.
23. SQL> grant dba to tcloud;
24. Grant succeeded.
25. SQL> conn tcloud/tcloud;
26. Connected.
27. SQL> create table t\_ogg(id int ,text\_name varchar(20),primary key(id));
28. Table created.

**目标端基础配置**

将下载到的对应OGG版本放在方便的位置并解压，本示例Oracle目标端最终的解压目录为/data/gg。

1. 配置环境变量  
   这里需要用到HDFS相关的库，故需要配置java环境变量以及OGG相关，并引入HDFS的相关库文件，参考配置如下：
2. export JAVA\_HOME=/usr/java/jdk1.7.0\_75/
3. export LD\_LIBRARY\_PATH=/usr/java/jdk1.7.0\_75/jre/lib/amd64:/usr/java/jdk1.7.0\_75/jre/lib/amd64/server:/usr/java/jdk1.7.0\_75/jre/lib/amd64/libjsig.so:/usr/java/jdk1.7.0\_75/jre/lib/amd64/server/libjvm.so:$OGG\_HOME:/lib
4. export OGG\_HOME=/data/gg
5. OGG初始化  
   目标端的OGG初始化和源端类似进入OGG的主目录执行./ggsci，进入OGG命令行
6. GGSCI (10.0.0.2) 2> create subdirs
7. Creating subdirectories under current directory /data/gg
8. Parameter files /data/gg/dirprm: already exists
9. Report files /data/gg/dirrpt: already exists
10. Checkpoint files /data/gg/dirchk: already exists
11. Process status files /data/gg/dirpcs: already exists
12. SQL script files /data/gg/dirsql: already exists
13. Database definitions files /data/gg/dirdef: already exists
14. Extract data files /data/gg/dirdat: already exists
15. Temporary files /data/gg/dirtmp: already exists
16. Credential store files /data/gg/dircrd: already exists
17. Masterkey wallet files /data/gg/dirwlt: already exists
18. Dump files /data/gg/dirdmp: already exists

**Oracle 源配置**

Oracle实时传输到Hadoop集群（HDFS，Hive，Kafka等）的基本原理如图：  
  
根据如上原理，配置大概分为如下步骤：源端目标端配置ogg管理器（mgr）；源端配置extract进程进行Oracle日志抓取；源端配置pump进程传输抓取内容到目标端；目标端配置replicate进程复制日志到Hadoop集群或者复制到用户自定义的解析器将最终结果落入到Hadoop集群。

**配置全局变量**

在源端服务器OGG主目录下，执行./ggsci到OGG命令行下，执行如下命令：

GGSCI (VM\_0\_25\_centos) 1> dblogin userid ggs password ggs

Successfully logged into database.

GGSCI (VM\_0\_25\_centos) 3> view params ./globals

ggschema ggs

其中./globals变量没有的话可以用edit params ./globals来编辑添加即可（编辑器默认使用的vim）

**配置管理器 mgr**

在OGG命令行下执行如下命令：

GGSCI (VM\_0\_25\_centos) 4> edit param mgr

PORT 7809

DYNAMICPORTLIST 7810-7909

AUTORESTART EXTRACT \*,RETRIES 5,WAITMINUTES 3

PURGEOLDEXTRACTS ./dirdat/\*,usecheckpoints, minkeepdays 3

说明：PORT即mgr的默认监听端口；DYNAMICPORTLIST动态端口列表，当指定的mgr端口不可用时，会在这个端口列表中选择一个，最大指定范围为256个；AUTORESTART重启参数设置表示重启所有EXTRACT进程，最多5次，每次间隔3分钟；PURGEOLDEXTRACTS即TRAIL文件的定期清理  
在命令行下执行start mgr即可启动管理进程，通过info mgr可查看mgr状态

GGSCI (VM\_0\_25\_centos) 5> info mgr

Manager is running (IP port VM\_0\_25\_centos.7809).

**添加复制表**

在OGG命令行下执行添加需要复制的表的操作，如下：

GGSCI (VM\_0\_25\_centos) 7> add trandata tcloud.t\_ogg

Logging of supplemental redo data enabled for table TCLOUD.T\_OGG.

GGSCI (VM\_0\_25\_centos) 8> info trandata tcloud.t\_ogg

Logging of supplemental redo log data is enabled for table TCLOUD.T\_OGG.

Columns supplementally logged for table TCLOUD.T\_OGG: ID.

**配置 extract 进程**

配置extract进程OGG命令行下执行如下命令：

GGSCI (VM\_0\_25\_centos) 10> edit params ext2hd

extract ext2hd

dynamicresolution

SETENV (ORACLE\_SID = "orcl")

SETENV (NLS\_LANG = "american\_america.AL32UTF8")

userid ggs,password ggs

exttrail /u01/gg/dirdat/tc

table tcloud.t\_ogg;

说明：第一行指定extract进程名称；dynamicresolution动态解析；SETENV设置环境变量，这里分别设置了Oracle数据库以及字符集；userid ggs,password ggs即OGG连接Oracle数据库的帐号密码，这里使用2.3.4中特意创建的复制帐号；exttrail定义trail文件的保存位置以及文件名，注意这里文件名只能是2个字母，其余部分OGG会补齐；table即复制表的表明，支持\*通配，必须以;结尾  
接下来在OGG命令行执行如下命令添加extract进程：

GGSCI (VM\_0\_25\_centos) 11> add extract ext2hd,tranlog,begin now

EXTRACT added.

最后添加trail文件的定义与extract进程绑定：

GGSCI (VM\_0\_25\_centos) 12> add exttrail /u01/gg/dirdat/tc,extract ext2hd

EXTTRAIL added

可在OGG命令行下通过info命令查看状态：

GGSCI (VM\_0\_25\_centos) 14> info ext2hd

EXTRACT EXT2HD Initialized 2016-11-09 15:37 Status STOPPED

Checkpoint Lag 00:00:00 (updated 00:02:32 ago)

Log Read Checkpoint Oracle Redo Logs

2016-11-09 15:37:14 Seqno 0, RBA 0

SCN 0.0 (0)

**配置 pump 进程**

pump进程本质上来说也是一个extract，只不过他的作用仅仅是把trail文件传递到目标端，配置过程和extract进程类似，只是逻辑上称之为pump进程  
在OGG命令行下执行：

GGSCI (VM\_0\_25\_centos) 16> edit params push2hd

extract push2hd

passthru

dynamicresolution

userid ggs,password ggs

rmthost 10.0.0.2 mgrport 7809

rmttrail /data/gg/dirdat/tc

table tcloud.t\_ogg;

说明：第一行指定extract进程名称；passthru即禁止OGG与Oracle交互，我们这里使用pump逻辑传输，故禁止即可；dynamicresolution动态解析；userid ggs,password ggs即OGG连接Oracle数据库的帐号密码，这里使用2.3.4中特意创建的复制帐号；rmthost和mgrhost即目标端OGG的mgr服务的地址以及监听端口；rmttrail即目标端trail文件存储位置以及名称  
分别将本地trail文件和目标端的trail文件绑定到extract进程：

GGSCI (VM\_0\_25\_centos) 17> add extract push2hd,exttrailsource /u01/gg/dirdat/tc

EXTRACT added.

GGSCI (VM\_0\_25\_centos) 18> add rmttrail /data/gg/dirdat/tc,extract push2hd

RMTTRAIL added.

同样可以在OGG命令行下使用info查看进程状态：

GGSCI (VM\_0\_25\_centos) 19> info push2hd

EXTRACT PUSH2HD Initialized 2016-11-09 15:52 Status STOPPED

Checkpoint Lag 00:00:00 (updated 00:01:04 ago)

Log Read Checkpoint File /u01/gg/dirdat/tc000000

First Record RBA 0

**配置 define 文件**

Oracle与MySQL，Hadoop集群（HDFS，Hive，kafka等）等之间数据传输可以定义为异构数据类型的传输，故需要定义表之间的关系映射，在OGG命令行执行：

GGSCI (VM\_0\_25\_centos) 20> edit params tcloud

defsfile /u01/gg/dirdef/tcloud.t\_ogg

userid ggs,password ggs

table tcloud.t\_ogg;

在OGG主目录下执行：  
./defgen paramfile dirprm/tcloud.prm  
完成之后会生成这样的文件/u01/gg/dirdef/tcloud.t\_ogg，将这个文件拷贝到目标端的OGG主目录下的dirdef目录即可。

**目标端的配置**

**创建目标表（目录）**

这里主要是当目标端为HDFS目录或者Hive表或者MySQL数据库时需要手动先在目标端创建好目录或者表，创建方法都类似，这里我们模拟实时传入到HDFS目录，故手动创建一个接收目录即可  
hadoop –fs mkdir /gg/replication/hive/

**配置管理器mgr**

目标端的OGG管理器（mgr）和源端的配置类似，在OGG命令行下执行：

GGSCI (10.0.0.2) 2> edit params mgr

PORT 7809

DYNAMICPORTLIST 7810-7909

AUTORESTART EXTRACT \*,RETRIES 5,WAITMINUTES 3

PURGEOLDEXTRACTS ./dirdat/\*,usecheckpoints, minkeepdays 3

**配置 checkpoint**

checkpoint即复制可追溯的一个偏移量记录，在全局配置里添加checkpoint表即可

GGSCI (10.0.0.2) 5> edit params ./GLOBALS

CHECKPOINTTABLE tcloud.checkpoint

保存即可

**配置 replicate 进程**

在OGG的命令行下执行：

GGSCI (10.0.0.2) 8> edit params r2hdfs

REPLICAT r2hdfs

sourcedefs /data/gg/dirdef/tcloud.t\_ogg

TARGETDB LIBFILE libggjava.so SET property=dirprm/hdfs.props

REPORTCOUNT EVERY 1 MINUTES, RATE

GROUPTRANSOPS 10000

MAP tcloud.t\_ogg, TARGET tcloud.t\_ogg;

说明：REPLICATE r2hdfs定义rep进程名称；sourcedefs即在3.6中在源服务器上做的表映射文件；TARGETDB LIBFILE即定义HDFS一些适配性的库文件以及配置文件，配置文件位于OGG主目录下的dirprm/hdfs.props；REPORTCOUNT即复制任务的报告生成频率；GROUPTRANSOPS为以事务传输时，事务合并的单位，减少IO操作；MAP即源端与目标端的映射关系  
其中property=dirprm/hdfs.props的配置中，最主要的几项配置及注释如下：

gg.handlerlist=hdfs //OGG for Big Data中handle类型

gg.handler.hdfs.type=hdfs //OGG for Big Data中HDFS目标

gg.handler.hdfs.rootFilePath=/gg/replication/hive/ //OGG for Big Data中HDFS存储主目录

gg.handler.hdfs.mode=op //OGG for Big Data中传输模式，即op为一次SQL传输一次，tx为一次事务传输一次

gg.handler.hdfs.format=delimitedtext //OGG for Big Data中文件传输格式

gg.classpath=/usr/hdp/2.2.0.0-2041/hadoop/share/hadoop/common/\*:/usr/hdp/2.2.0.0-2041/hadoop/share/hadoop/common/lib/\*:/usr/hdp/2.2.0.0-2041/hadoop/share/hadoop/hdfs/\*:/usr/hdp/2.2.0.0-2041/hadoop/etc/hadoop/:/data/gg/:/data/gg/lib/\*:/usr/hdp/2.2.0.0-2041/hadoop/client/\* //OGG for Big Data中使用到的HDFS库的定义

具体的OGG for Big Data支持参数以及定义可参考[地址](http://docs.oracle.com/goldengate/bd1221/gg-bd/GADBD/GUID-85A82B2E-CD51-463A-8674-3D686C3C0EC0.htm" \l "GADBD376" \t "_blank)

最后在OGG的命令行下执行：

GGSCI (10.0.0.2) 9> add replicat r2hdfs exttrail /data/gg/dirdat/tc,checkpointtable tcloud.checkpointtab

REPLICAT added.

将文件与复制进程绑定即可

**测试**

**启动进程**

在源端和目标端的OGG命令行下使用start [进程名]的形式启动所有进程。

启动顺序按照源mgr——目标mgr——源extract——源pump——目标replicate来完成。

**检查进程状态**

以上启动完成之后，可在源端与目标端的OGG命令行下使用info [进程名]来查看所有进程状态，如下：  
源端：

GGSCI (VM\_0\_25\_centos) 7> info mgr

Manager is running (IP port VM\_0\_25\_centos.7809).

GGSCI (VM\_0\_25\_centos) 9> info ext2hd

EXTRACT EXT2HD Last Started 2016-11-09 16:05 Status RUNNING

Checkpoint Lag 00:00:00 (updated 00:00:09 ago)

Log Read Checkpoint Oracle Redo Logs

2016-11-09 16:45:51 Seqno 8, RBA 132864000

SCN 0.1452333 (1452333)

GGSCI (VM\_0\_25\_centos) 10> info push2hd

EXTRACT PUSH2HD Last Started 2016-11-09 16:05 Status RUNNING

Checkpoint Lag 00:00:00 (updated 00:00:01 ago)

Log Read Checkpoint File /u01/gg/dirdat/tc000000

First Record RBA 1043

目标端：

GGSCI (10.0.0.2) 13> info mgr

Manager is running (IP port 10.0.0.2.7809, Process ID 8242).

GGSCI (10.0.0.2) 14> info r2hdfs

REPLICAT R2HDFS Last Started 2016-11-09 16:45 Status RUNNING

Checkpoint Lag 00:00:00 (updated 00:00:02 ago)

Process ID 4733

Log Read Checkpoint File /data/gg/dirdat/tc000000

First Record RBA 0

所有的状态均是RUNNING即可。（当然也可以使用info all来查看所有进程状态）

**测试同步更新效果**

测试方法比较简单，直接在源端的数据表中insert，update，delete操作即可。由于Oracle到Hadoop集群的同步是异构形式，目前尚不支持truncate操作。  
源端进行insert操作

SQL> conn tcloud/tcloud

Connected.

SQL> select \* from t\_ogg;

no rows selected

SQL> desc t\_ogg;

Name Null? Type

----------------------------------------- -------- ----------------------------

ID NOT NULL NUMBER(38)

TEXT\_NAME VARCHAR2(20)

SQL> insert into t\_ogg values(1,'test');

1 row created.

SQL> commit;

Commit complete.

查看源端trail文件状态

[oracle@VM\_0\_25\_centos dirdat]$ ls -l /u01/gg/dirdat/tc\*

-rw-rw-rw- 1 oracle oinstall 1180 Nov 9 17:05 /u01/gg/dirdat/tc000000

查看目标端trail文件状态

[root@10 dirdat]# ls -l /data/gg/dirdat/tc\*

-rw-r----- 1 root root 1217 Nov 9 17:05 /data/gg/dirdat/tc000000

查看HDFS中是否有写入

hadoop fs -ls /gg/replication/hive/tcloud.t\_ogg

-rw-rw-r-- 3 root hdfs 110 2016-11-09 17:05

/gg/replication/hive/tcloud.t\_ogg/tcloud.t\_ogg\_2016-11-09\_17-05-30.514.txt

注意：从写入到HDFS的文件内容看，文件的格式如下：

ITCLOUD.T\_OGG2016-11-09 09:05:25.0670822016-11-09T17:05:30.51200000000000000000001080ID1TEXT\_NAMEtest

很明显Oracle的数据已准实时导入到HDFS了。导入的内容实际是一条条的类似流水日志（具体日志格式不同的传输格式，内容略有差异，本例使用的delimitedtext。格式为操作符 数据库.表名 操作时间戳(GMT+0) 当前时间戳(GMT+8) 偏移量 字段1名称 字段1内容 字段2名称 字段2内容），如果要和Oracle的表内容完全一致，需要客户手动实现解析日志并写入到Hive的功能，这里官方并没有提供适配器。目前腾讯侧已实现该功能的开发。  
当然你可以直接把这个HDFS的路径通过LOCATION的方式在Hive上建外表（external table）达到实时导入Hive的目的。

**总结**

OGG for Big Data实现了Oracle实时同步到Hadoop体系的接口，但得到的日志目前仍需应用层来解析（关系型数据库如MySQL时OGG对应版本已实现应用层的解析，无需人工解析）。  
OGG的几个主要进程mgr，extract，pump，replicate配置方便，可快速配置OGG与异构关系存储结构的实时同步。后续如果有新增表，修改对应的extract，pump和replicate进程即可，当然如果是一整个库，在配置上述2个进程时，使用通配的方式即可。

**附录**

OGG到Hadoop体系的实时同步时，可在源端extract和pump进程配置不变的情况下，直接在目标端增加replicate进程的方式，增加同步目标，以下简单介绍本示例中增加同步到Kafka的配置方法。  
本示例中extract，pump进程都是现成的，无需再添加。只需要在目标端增加同步到Kafka的replicate进程即可。

在OGG的命令行下执行：

GGSCI (10.0.0.2) 4> edit params r2kafka

REPLICAT r2kafka

sourcedefs /data/gg/dirdef/tcloud.t\_ogg

TARGETDB LIBFILE libggjava.so SET property=dirprm/r2kafka.props

REPORTCOUNT EVERY 1 MINUTES, RATE

GROUPTRANSOPS 10000

MAP tcloud.t\_ogg, TARGET tcloud.t\_ogg;

replicate进程和导入到HDFS的配置类似，差异是调用不同的配置dirprm/r2kafka.props。这个配置的主要配置如下：

gg.handlerlist = kafkahandler //handler类型

gg.handler.kafkahandler.type = kafka

gg.handler.kafkahandler.KafkaProducerConfigFile=custom\_kafka\_producer.properties //kafka相关配置

gg.handler.kafkahandler.TopicName =ggtopic //kafka的topic名称，无需手动创建

gg.handler.kafkahandler.format =json //传输文件的格式，支持json，xml等

gg.handler.kafkahandler.mode =op //OGG for Big Data中传输模式，即op为一次SQL传输一次，tx为一次事务传输一次

gg.classpath=dirprm/:/usr/hdp/2.2.0.0-2041/kafka/libs/\*:/data/gg/:/data/gg/lib/\* //相关库文件的引用

r2kafka.props引用的custom\_kafka\_producer.properties定义了Kafka的相关配置如下：

bootstrap.servers=10.0.0.62:6667 //kafkabroker的地址

acks=1

compression.type=gzip //压缩类型

reconnect.backoff.ms=1000 //重连延时

value.serializer=org.apache.kafka.common.serialization.ByteArraySerializer

key.serializer=org.apache.kafka.common.serialization.ByteArraySerializer

batch.size=102400

linger.ms=10000

以上配置以及其他可配置项可参考[地址](http://docs.oracle.com/goldengate/bd1221/gg-bd/GADBD/GUID-2561CA12-9BAC-454B-A2E3-2D36C5C60EE5.htm" \l "GADBD449" \t "_blank)：

以上配置完成后，在OGG命令行下添加trail文件到replicate进程并启动导入到Kafka的replicate进程

GGSCI (10.0.0.2) 5> add replicat r2kafka exttrail

/data/gg/dirdat/tc,checkpointtable tcloud.checkpoint

REPLICAT added.

GGSCI (10.0.0.2) 6> start r2kafka

Sending START request to MANAGER ...

REPLICAT R2KAFKA starting

GGSCI (10.0.0.2) 10> info r2kafka

REPLICAT R2KAFKA Last Started 2016-11-09 17:59 Status RUNNING

Checkpoint Lag 00:00:00 (updated 00:00:09 ago)

Process ID 5236

Log Read Checkpoint File /data/gg/dirdat/tc000000

2016-11-09 17:05:25.067082 RBA 1217

检查实时同步到kafka的效果，在Oracle源端更新表的同时，使用kafka客户端自带的脚本去查看这里配置的ggtopic这个kafkatopic下的消息：

SQL> insert into t\_ogg values(2,'test2');

1 row created.

SQL> commit;

Commit complete.

目标端Kafka的同步情况：

[root@10 kafka]# bin/kafka-console-consumer.sh --zookeeper 10.0.0.223:2181 --

from-beginning --topic ggtopic

{"table":"TCLOUD.T\_OGG","op\_type":"I","op\_ts":"2016-11-09

09:05:25.067082","current\_ts":"2016-11-

09T17:59:20.943000","pos":"00000000000000001080","after":

{"ID":"1","TEXT\_NAME":"test"}}

{"table":"TCLOUD.T\_OGG","op\_type":"I","op\_ts":"2016-11-09

10:02:06.827204","current\_ts":"2016-11-

09T18:02:12.323000","pos":"00000000000000001217","after":

{"ID":"2","TEXT\_NAME":"test2"}}

显然，Oracle的数据已准实时同步到Kafka。从头开始消费这个topic发现之前的同步信息也存在。架构上可以直接接Storm，SparkStreaming等直接消费kafka消息进行业务逻辑的处理。  
从Oracle实时同步到其他的Hadoop集群中，官方最新版本提供了HDFS，HBase，Flume和Kafka，相关配置可参考官网给出的例子配置即可。

<https://www.pythian.com/blog/goldengate-12-2-big-data-adapters-part-5-mongodb/>

https://www.qcloud.com/community/article/164816001481011932