



集算器

高性能计算专家

OGG增量采集数据入库

润乾软件出品

主要内容



- 1、OGG数据入库前言
- 2、数据采集流程
- 3、目录与数据文件
- 4、OGG集成JAVA插件
- 5、数据同步操作
- 6、后台程序合并数据
- 7、用SPL数据入库
- 8、最后总结



1、OGG数据入库前言

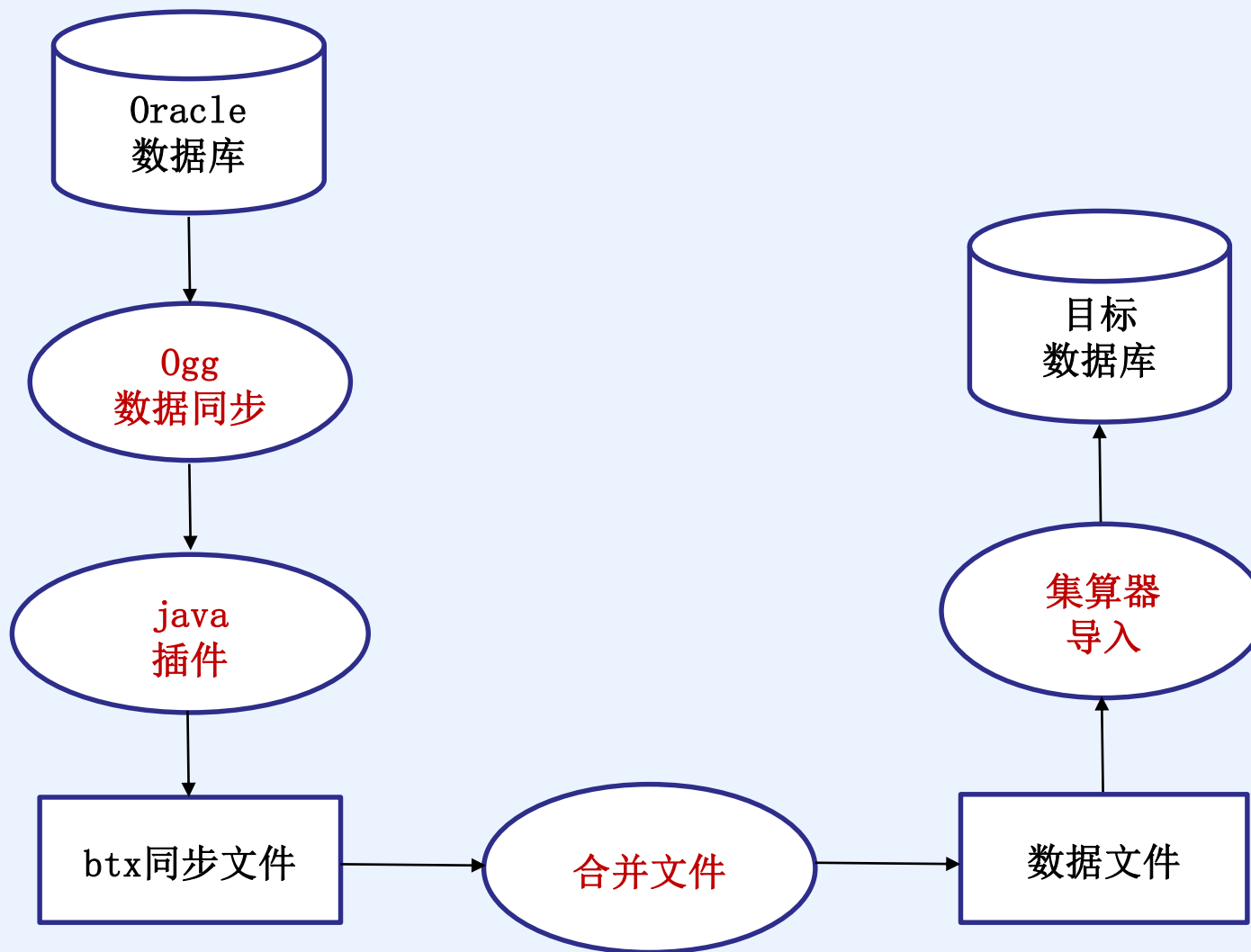
Oracle Golden Gate (简称ogg) 软件是一种基于日志的结构化数据复制备份软件，它通过解析源数据库日志获得数据的增量变化，再将这些变化应用到目标数据库，从而实现源数据库与目标数据库同步。

ogg可以灵活地在同类和异类系统（包括不同版本的 Oracle Database、不同的硬件平台）之间以及 Oracle 数据库和非 Oracle 数据库（包括MSSQL、IBM DB2、Mysql, MongoDB等）之间移动数据, 可根据目标数据库的需要，复制所有数据或部分数据。

面对ogg支持或不支持的目标数据库，ogg的配置不尽相同。但通过SPL语言导入不同的数据库，能很好地实现统一接口，方便在不同目的数据库之间进行切换。

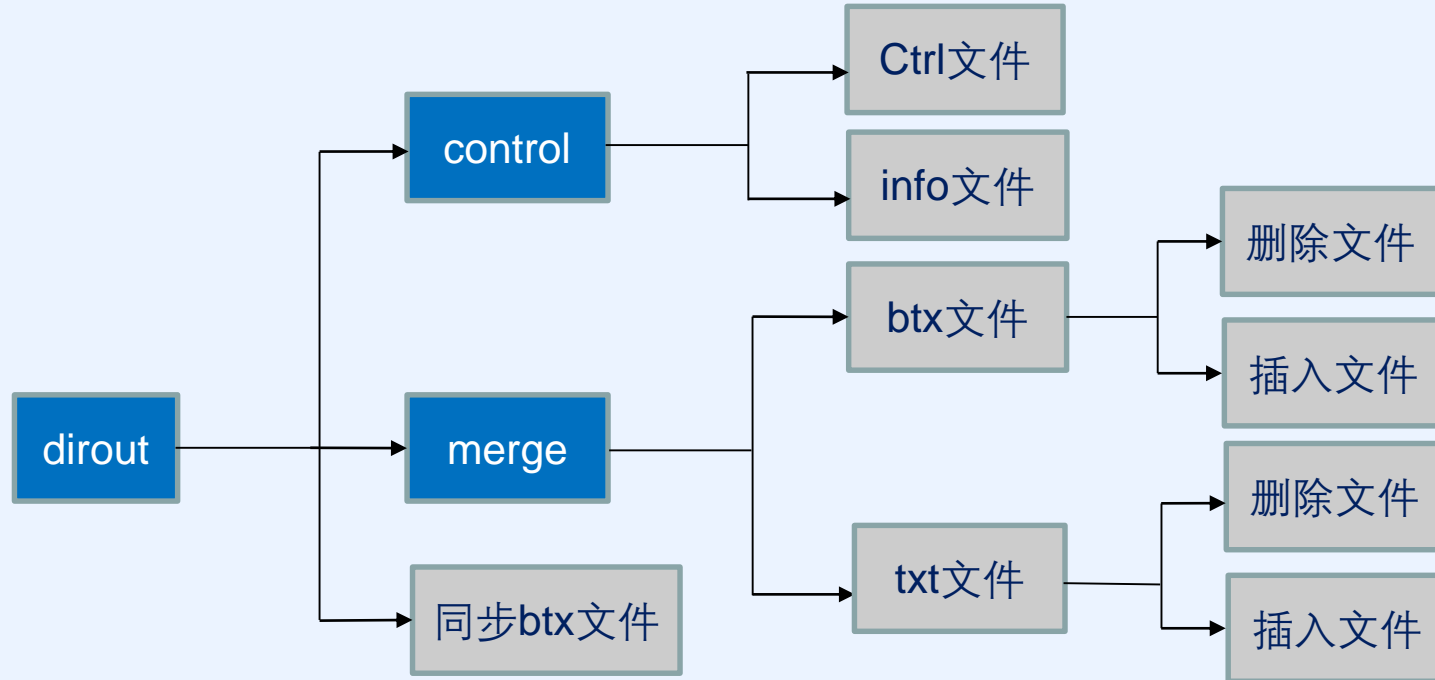
本解决方案通过java插件，将ogg同步的数据归纳合并为删除与插入数据，结合集算器SPL语言支持众多的关系数据库与非关系数据库，以方便简化数据入库操作处理，支持oracle向不同数据库进行数据同步。

2、数据采集流程图





3、目录与数据文件



dirout、control、meger为目录，其它为文件。

control目录：存放info信息文件与ctrl索引文件

merge目录：存放合并后的btx数据文件，txt数据文件。

同步btx文件：记录从oracel同步过来数据，入库前需要合并处理。

合并文件：merge目录下的btx，txt文件，经过合并处理后将生成删除操作文件、插入操作文件，作为入库数据用。

文件名称命名规则



类型	命名规则	样例
同步btx文件	tablespace_tablename_date_time_sn.btx	OGG_DEMO_2019-06-26_14-10-55_00002.btx
合并文件	PUMP_tablespace_tablename_op_datetime.btx	PUMP_OGG_KULL_I_2019-08-30_19-02-58.btx
info信息文件	PK_tablespace_tablename_info.txt	PK_OGG_TT_info.txt
索引文件	tablespace_tablename_date.ctrl	OGG_TT_2019-08-29_data.ctrl

文件结构



info信息文件

记录表结构信息的文件，包括表名、字段名、字段类型及主键。

```
1 ID;NAME;GRADE
2 DOUBLE;VARCHAR;DOUBLE
3 PK=1;ID
4
```

```
1 PID;NAME;GRADER
2 DOUBLE;VARCHAR;DOUBLE
3 PK=0;PID;NAME;GRADER
4
```

内容共三行，分别为字段名、字段类型及主键字段。

其中PK=1表示有主键，后面记录对应的字段名；

PK=0则表示无主键，后面记录表所有的字段。



索引文件

记录某表某天保存的btx文件名，当执行SQL DML操作，commit提交数据后，将生成对应的btx数据文件，索引文件则会记录这些按时间先后顺序排列的文件名，以便合并btx文件时用

```
0 10 20 30 40
1 dirout/OGG_TT_2019-08-29_15-52-14_00000.btx
2 dirout/OGG_TT_2019-08-29_15-55-02_00000.btx
3 dirout/OGG_TT_2019-08-29_16-08-46_00000.btx
4 dirout/OGG_TT_2019-08-29_16-10-56_00000.btx
5 dirout/OGG_TT_2019-08-29_17-10-00_00000.btx
6 dirout/OGG_TT_2019-08-29_17-48-41_00000.btx
7 dirout/OGG_TT_2019-08-29_17-52-41_00000.btx
8
```


文件结构



数据文件

存放在merge下的txt文件、btx文件，分别为文本格式与二进制格式，它们的数据结构相同，均为事件类型+字段+数据形式

```
0 10 20
1 OP;ID;NAME;GRADE;
2 I;2111;cc1101;110;
3 I;2128;cc1102;230;
4 I;2129;cc1103;260;
5
```

```
0 10 20
1 OP;ID;NAME;GRADE;
2 D;2114;null;null;
3 D;2116;null;null;
4 D;2117;null;null;
5
```

OP表示事件类型，其值为I或D，分别表示插入事件与删除事件。

首行为信息数据，OP事件类型后面跟字段名称，非首行是数据记录。

在删除文件中，若数据库表有主键，则非主键字段对应的数据为空。



4、OGG集成JAVA插件

在OGG Adapter Java基础上开发的java插件，主要实现下面接口，生成oggplug.jar文件。

```
package com.raqsoft.lib.ogg;

public class SplHandler extends AbstractHandler {
    public void init(DsConfiguration conf, DsMetaData metaData);
    public Status transactionCommit(DsEvent e, DsTransaction tx);
    public Status operationAdded(DsEvent e, DsTransaction tx, DsOperation dsOperation);
    public void destroy();
}
```



4、OGG集成JAVA插件

javaue.properties配置文件

```
1 gg.handlerlist=raq
2 gg.handler.raq.type=com.raqsoft.lib.ogg.SplHandler
3 goldengate.userexit.timestamp=utc
4 goldengate.userexit.nockpt=true
5 goldengate.userexit.writers=javawriter
6
7 goldengate.log.logname=cuserexit
8 goldengate.log.level=INFO
9 goldengate.log.tofile=true
10
11 javawriter.stats.display=TRUE
12 javawriter.stats.full=TRUE
13
14 javawriter.bootoptions=-Djava.class.path=.;dirprm;ggjava/resources/classes;ggjava/resou
.rces/lib;ggjava/ggjava.jar;dirprm/fastjson-1.2.2.jar;dirprm/oggplug.jar;dirprm/icu4j_3_
.4_5.jar;dirprm/dm.jar -Dlog4j.configuration=log4j.properties -Ddebug.trc=true
```

gg.handler.raq.type指向SplHandler类，依赖包中增加oggplug.jar。



4、OGG集成JAVA插件

Javaue. rpm配置文件

Extract JAVAUE

-- the source-def's must match the trail data

SourceDefs dirdef/ogg.def

-- windows:

CUserExit ggjava_ue.dll CUSEREXIT PassThru IncludeUpdateBefore

-- unix/linux:

--CUserExit libggjava_ue.so CUSEREXIT PassThru IncludeUpdateBefore

GetUpdateBefore

Table OGG.*;

设置连接数据源SourceDefs及依赖库等



5、数据同步操作

下面通过具体的表数据操作来说明数据同步。

A. 数据库表结构：

无主键的表KULL结构如下：

字段	类型	NULL	注释
PID	NUMBER	N	记录ID
NAME	VARCHAR2 (20)	Y	名称
GRADE	INTEGER	Y	分数

带主键的表test结构如下：

字段	类型	NULL	注释
PNO	INTEGER	N	批次号
SNO	INTEGER	N	流水号，与PNO组成主键
NAME	VARCHAR2 (20)	Y	名称
VAL	VARCHAR2 (256)	Y	参数说明



5、数据同步操作

数据操作前KULL表数据:

	PID	NAME	GRADE
1	3126	mm1000	1000
2	3127	cc1107	3700

数据操作前TEST表数据:

	PNO	SNO	NAME	VAL
1	200	200	Am202	Bm202
2	300	100	Am301	Bm301
3	300	200	Am100	Bm100



A. 单表数据操作: delete

	PID	NAME	GRADE
1	3126	mm1000	1000
2	3127	cc1107	3700



```
savefile :dirout/OGG_KULL_2019-08-30_13-14-43_00015. btx
OP      PID      NAME      GRADER
D       3126     mm1000    1000
D       3127     cc1107    3700
transactionCommit xxxxxxxxxxxxxxxxxxxxxxx
```



A. 单表数据操作: insert

	PID	NAME	GRADE
1	3126	cc1106	3600
2	3127	cc1107	3700



```
savefile :dirout/OGG_KULL_2019-08-30_13-14-43_00018.btx
OP      PID      NAME      GRADER
I       3126    cc1106    3600
I       3127    cc1107    3700
transactionCommit xxxxxxxxxxxxxxxxxxxxxxxx
```




A. 单表数据操作: update

	PID	NAME	GRADE
更新前	3126	cc1106	3600
更新后	3126	mm1000	1000



```
savefile :dirout/OGG_KULL_2019-08-30_13-14-43_00021.btx
OP      PID      NAME      GRADER
D       3126     cc1106    3600
I       3126     mm1000    1000
transactionCommit xxxxxxxxxxxxxxxxxxxxxxx
```



A. 单表数据DML操作

数据同步说明

※ 对数据进行DML操作，每次commit提交后，ogg根据数据库表生成对应的同步btx文件，文件保存变化的数据(增删改事件的数据)。

※ OP=D表示删除，OP=I表示插入，update操作被分解成删除与插入操作



B. 多张表同时操作举例说明

执行delete, insert, update的sql语句:

```
delete from KULL;
insert into KULL values(3126, 'cc1106', 3600);
insert into KULL values(3127, 'cc1107', 3700);
update KULL set GRADER=1000,name='mm1000' where PID = 3126;

delete from test;
insert into test values(200, 200, 'Am202', 'Bm202');
insert into test values(300, 100, 'Am301', 'Bm301');
insert into test values(300, 200, 'Am302', 'Bm302');
update test set name='Am100',val='Bm100' where pno=300 and sno=200;
commit;
```



B. 多张表同时操作举例说明

执行delete, insert, update sql语句:

oracle数据

表ku11

	PID	NAME	GRADER
▶ 1	3127	cc1107	3700
2	3126	cc1106	3600

表test

	PNO	SNO	NAME	VAL
▶ 1	200	200	Am202	Bm202
2	300	100	Am301	Bm301
3	300	200	Am100	Bm100

同步btx数据

```
savefile :dirout/OGG_KULL_2019-08-30_11-33-40_00012.btx
OP      PID      NAME      GRADER
D       3126     mm1000    1000
D       3127     cc1107    3700
I       3126     cc1106    3600
I       3127     cc1107    3700
D       3126     cc1106    3600
I       3126     mm1000    1000
```

```
savefile :dirout/OGG_TEST_2019-08-30_11-33-40_00014.btx
OP      PNO      SNO      NAME      VAL
D       200      200      null      null
D       300      100      null      null
D       300      200      null      null
I       200      200      Am202     Bm202
I       300      100      Am301     Bm301
I       300      200      Am302     Bm302
D       300      200      null      null
I       300      200      Am100     Bm100
```



B.多张表同时操作举例说明

数据同步说明

- A、表KULL数据变化与前面类似，表test删除时略有不同。
- B、由于test带主键，在删除操作时，非主键字段对应值为空。
- C、SQL语句中的数据记录按表名分别记录到对应的同步btx文件中。
- D、记录的顺序与操作顺序保持一致。

由于SQL操作过程中用commit事务提交后，就会产生同步记录文件，同步的btx文件可能会比较多，尤其是对多表频繁操作情况下。因此后面需要进行文件合并，为方便入库处理作准备。



6、后台程序合并数据

※ 合并数据指同步后的btx文件数据，在某个时间段内按不同的表名进行多个文件数据合并，并根据数据操作的先后顺序进行处理，减少过多的数据文件问题。

※ 合并完成后，每张表会生成删除事件数据文件、插入事件数据文件(若对应的事件存在)，被处理过的同步btx文件会被删除。

※ 合并即包括文件上的合并，也包括数据记录上的合并。

※ 合并程序merge.bat在后台运行，参数缺省情况下每小时自动执行合并一次，生成的文件存放在dirout/merge目录下。

※ 启动后台程序：

```
>merge.bat ./
```



以无主键表kull合并为例来说明

A、文件合并

```
dirout/OGG_KULL_2019-08-30_13-14-43_00015.btx  
dirout/OGG_KULL_2019-08-30_13-14-43_00018.btx  
dirout/OGG_KULL_2019-08-30_13-14-43_00021.btx
```

合并处理

```
dirout/merge/PUMP_OGG_KULL_D_2019-08-30_14-38-33.btx  
dirout/merge/PUMP_OGG_KULL_I_2019-08-30_14-38-33.btx
```



以无主键表kull合并为例来说明

B、数据合并

同步btx数据

OP	PID	NAME	GRADER
D	3126	mm1000	1000
D	3127	cc1107	3700
I	3126	cc1106	3600
I	3127	cc1107	3700
D	3126	cc1106	3600
I	3126	mm1000	1000

合并处理

合并后数据

序号	OP	PID	NAME	GRADER
1	<u>D</u>	<u>3126</u>	<u>cc1106</u>	<u>3600</u>
2	<u>D</u>	<u>3127</u>	<u>cc1107</u>	<u>3700</u>
3	<u>D</u>	<u>3126</u>	<u>mm1000</u>	<u>1000</u>

序号	OP	PID	NAME	GRADER
1	<u>I</u>	<u>3127</u>	<u>cc1107</u>	<u>3700</u>
2	<u>I</u>	<u>3126</u>	<u>mm1000</u>	<u>1000</u>

对于同一条记录执行先插入后删除处理，则不再记录其插入数据。如插入记录(3126, 'cc1106', 3600)



以有主键表test合并为例来说明

A、文件合并

dirout/OGG_TEST_2019-08-30_11-33-40_00014.btx



合并处理

dirout/merge/PUMP_OGG_TEST_D_2019-08-30_15-07-54.btx
dirout/merge/PUMP_OGG_TEST_I_2019-08-30_15-07-54.btx



以有主键表test合并为例来说明

B、数据合并

同步btx数据

OP	PNO	SNO	NAME	VAL
D	200	200	null	null
D	300	100	null	null
D	300	200	null	null
I	200	200	Am202	Bm202
I	300	100	Am301	Bm301
I	300	200	Am302	Bm302
D	300	200	null	null
I	300	200	Am100	Bm100

合并处理

合并后数据

序号	OP	PNO	SNO	NAME	VAL
1	<u>D</u>	<u>300</u>	<u>200</u>	(null)	(null)
2	<u>D</u>	<u>300</u>	<u>100</u>	(null)	(null)
3	<u>D</u>	<u>200</u>	<u>200</u>	(null)	(null)

序号	OP	PNO	SNO	NAME	VAL
1	<u>I</u>	<u>200</u>	<u>200</u>	<u>Am202</u>	<u>Bm202</u>
2	<u>I</u>	<u>300</u>	<u>100</u>	<u>Am301</u>	<u>Bm301</u>
3	<u>I</u>	<u>300</u>	<u>200</u>	<u>Am100</u>	<u>Bm100</u>

1. 由于带主键，删除时只需要PNO, SNO字段，其它字段值可忽略。
2. 对于删除记录，有重复的记录，则先去重后再存储。插入记录中有重复数据也如此。



Merge合并程序介绍

```
*****  
* Usage: merge.bat path key:value,....  
* key: model, datetime, interval, filetype  
* 1. model:使用模式,分为自动处理auto,手动处理manual、缺省为自动auto;  
* 2. datetime:手动处理合并文件的起始时间,时间格式为yyyy-MM-dd HH:00:00  
* 3. interval:自动处理的间隔时间,缺省为60分钟(单位为分钟);  
* 4. filetype:输出文件格式,分为txt、btx文件格式,缺省为btx  
* Auto Example:  
* merge.bat ./ "model:auto,datetime:2019-07-05 15:00:00,interval:5,filetype:txt"  
*****/
```

model: 自动还是手动处理。若为手动,则interval间隔时间参数无效,它会立即执行操作;若为自动,则datetime参数无效,它将定时执行操作。

datetime: 自定义合并时间,以整点为开始的1个小时时间段。如过去有段时间的数据没有合并,则在手动模式下可用此参数。

interval: 间隔时间,单位为分钟,自动合并模式下,每隔给定时长循环执行合并。

filetype: 输出文件类型,支持二进制格式btx文件,文本格式txt文件。



7、用SPL数据入库

将bt_x文件数据导入数据库，基本流程是加载数据后，先进行删除操作，再进行插入操作。
使用集算器SPL的代码如下：

	A	B
1	=connect("mysql")	
2	=A1.query("select * from kull").keys(PID)	
3	=file("D:/app/orcl/product/ggs/dirout/merge/ PUMP_OGG_KULL_D_2019-08-30_14-38-33.bt _x ").import@b()	
4	=file("D:/app/orcl/product/ggs/dirout/merge/ PUMP_OGG_KULL_I_2019-08-30_14-38-33.bt _x ").import@b()	
5	=A4.run(PID=int(PID))	
6	"delete from kull where "	
7	for A3	"PID="+A7.PID+" and " + "NAME="+A7.NAME+" and " + "GRADER="+A7.GRADER
8		>A1.execute(A6+B7)
9	=mysql.update@i(A4:A2,kull, PID, NAME, GENDER; PID)	
10	=A1.query("select * from kull")	
11	>A1.close()	



7、用SPL数据入库

执行后结果：

序号	PID	NAME	GRADER
1	2127	<u>cc1107</u>	2700
2	2129	<u>cc1109</u>	2900
3	3127	<u>cc1107</u>	3700
4	3126	<u>mm1000</u>	1000

前两条为数据库原有的数据，后面两条为同步的数据。

表test入库操作的SPL脚本类似，在此略过。

合并后的btX文件数据可根据实际需求，可导入不同类型的数据库。

已经处理过的btX或txt文件处理办法：

确认入库数据正确无误后，可删除btX或txt文件，也可移动到其它位置作备份，处理时注意避免数据重复入库。



7、用SPL数据入库

针对同一张表入库操作说明：

1. 每个btx文件是相对独立的，它本身是记录一段时间内的数据变化。
2. 每次SPL入库处理是一样的，传递要处理的文件名作为参数则可以复用SPL脚本。
3. 入库操作执行顺序是先删除后插入操作，删除事件数据对数据库中已存在数据有影响，同样对插入数据也有影响，若不先执行删除操作，可能会出现主键重复问题。
4. 对多个btx文件入库操作，就时间先后顺序而言，生成文件时间在前的先执行，时间在后的后执行，这样与DML数据操作顺序保持一致。

在入库处理时，对于txt格式数据文件，也可以结合info文件表信息，用其它工具来实现。自动入库处理，可将入库处理程序加入定时器中执行。检测不到dirout/merging.lck文件时，表明merge程序没有在进行合并处理(或者说已经处理完)，此时可执行入库操作。



8、最后总结

ogg配置后，启动两端各进程及合并后台程序，对数据DML操作后，ogg对数据自动采集生成同步btx文件，后台程序合并btx文件后，通过SPL进行入库处理。在了解ogg 数据采集原理与操作过程后，用户要做的是如何将合并后的数据导入数据库中。

简化流程图如下：

