# 元数据调试指导手册

## 1、任务调试及问题定位

#### 1.1、登陆MySQL查看任务执行情况

用putty或者x-shell等工具登陆MySQL所在的服务器，查看任务执行情况。这里提供一个MySQL可视化的查询工具。

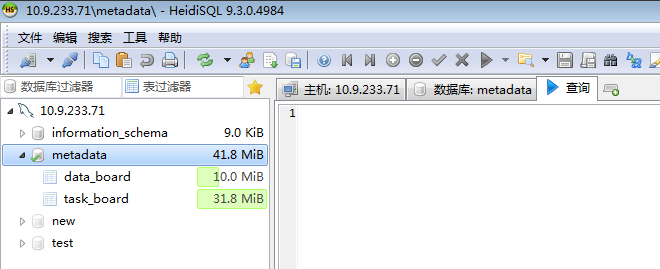
**查询工具:**



1、登陆MySQL：用户名：zxvmax，密码：zxpos，端口：3310



2、切换数据库metadata，编辑执行查询语句



3、查看数据公告牌和任务公告牌字段

**desc** task\_board, **desc** data\_board

4、执行查询语句，常用的如下：

1）查看所有任务执行情况

**select** resultstatus ,**count**(\*) **from** task\_board **where** **day**='2016-04-06' **group** **by** resultstatus;

2）查看具体某个功能的任务执行情况

**select** \* **from** task\_board **where** virtualtaskname =’cdma\_topn\_task’ **and** **day**='2016-04-06' **order** **by** taskname;

3）查看某个任务的执行情况

**select** \* **from** task\_board **where** taskname='cdma\_netmaxc\_subject\_topn\_1x\_day' **and** **day**='2016-04-06'

4）查看所有失败的任务

**select** **distinct** taskname,virtualtaskname **from** task\_board **where** resultstatus **like** 'submitted%'

#### 1.2、找出失败任务，定位失败原因

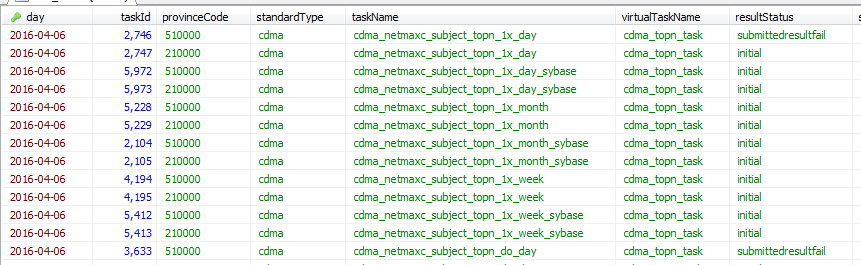
**第一步**：执行1.1节提到的任务状态查询语句，找出失败的任务，记录任务的下发的时间day、taskname、taskid，失败的任务包括submittedresultfail、submittedfail、submittedterminated三种状态。

Submittedresultfail：任务在执行由于自身或者环境原因失败，这类失败的任务需要开发人员定位问题的原因。

Submittedfail、submittedterminated：任务在提交到spark执行过程中，由于线程原因导致失败，这类失败任务一般先可以忽略，等待任务重新提交执行（默认会重跑三次）。

**示例：**以topN功能为例：执行查询语句。

**select** \* **from** task\_board **where** virtualtaskname ='cdma\_topn\_task' **and** **day**='2016-04-06' **order** **by** taskname;



**第二步：**如果任务处于Submittedresultfail，需要找出具体任务执行的log，查看任务的执行情况。

Log路径位于：

/home/netnumen/ems/ums-server/utils/vmax-metadata-manager/submitSparkAppFile/cdma\_netmaxc\_subject\_topn\_1x\_day

通过任务公告牌对应任务的day、taskname、taskid，找到对应任务的log文件。



**示例：**以cdma\_netmaxc\_subject\_topn\_1x\_day任务为例，执行log为：

meta\_result\_2016004007\_2746\_cdma\_netmaxc\_subject\_topn\_1x\_day\_1\_vmax.json.log

打开log查看任务执行情况。

**第三步**： sql算法任务，可以通过验证log中打印的sql找出问题原因。一般sql任务的失败都是因为算法语法有问题。报错类型为：

1. ERROR SqlAction: 算法解析出错，xml中输入输出参数配置有问题。
2. Sql语法本身有问题
3. 常见问题参见2.1

**第四步：**rdd形式的任务，建议在rdd架包中的关键位置多加一些打印，这样便于快速定位问题原因。这类任务的失败，大都是因为架包打的不对或者rdd中的代码本身有问题。报错类型为：

1、class not find,找不到类：rdd架包打包有问题或者xml配置的jarpath不对。

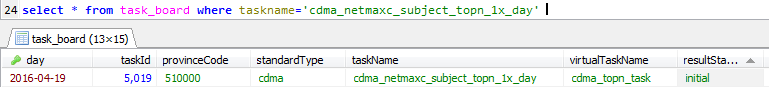
2、rdd架包内的代码本身有问题。

3、常见问题参见2.1

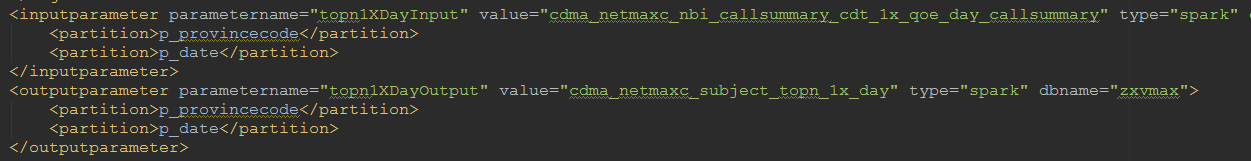
#### 1.3、找出initial状态任务，分析未执行原因

initial状态任务定位的原则是：找到数据源最开始的地方，看看有没有数据。任务能否执行，主要看它依赖的表的数据有没有到，数据公告牌有没有翻牌。是否翻牌由data\_board中的state字段标示，如果state=1，表示数据到达、公告牌翻牌；state=0，表示无数据，任务不会执行。因此，对于一个initial状态的任务，可以通过查看数据有没有，快速定位原因。

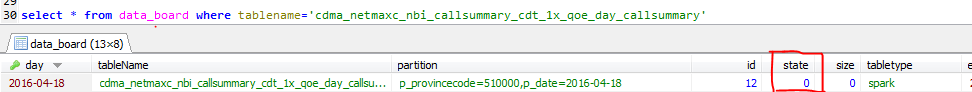
**示例：**以cdma\_netmaxc\_subject\_topn\_1x\_day任务为例，



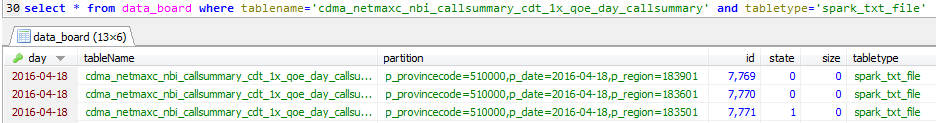
**第一步**：检查initial任务的xml的配置，查看任务依赖的输入表有哪些。



**第二步**：查看依赖的输入表有没有数据，数据公告牌是否翻牌。



**第三步**：查看表的入库数据有没有到达，数据公告牌是否翻牌



如上图所示，入库任务的表只到了一个region的数据，其他没有到齐，导致入库的任务不会执行，相应的原始表的数据公告牌也不会翻牌，从而影响清洗任务的完成。

**第四步**：如入库任务依赖的表未翻牌，则去相应的hdfs路径查看，数据有没有推送过来。如无数据，则需要查看drs的配置的数据源。

此外，任务未执行也可能是因为它依赖的某个主题表清洗失败，导致数据公告牌没翻牌。如天依赖小时，周月又依赖天，如果24个小时任务有一个失败，都会影响后续任务的执行。

总之，只要把握initial状态任务定位的原则，一个个往上追溯，直至找到源头，大都能找到原因。某些特殊情况，如配置了延时翻牌，则需要视具体情况而定。

#### 1.4、sybase同步任务验证

需要明确的是，我们在数据库中看到的sybase任务执行成功，并意味着数据已经同步到sybase中。

**1、Sybase同步任务元数据实现机制**：首先，执行我们配置的sybase任务将文件下载到相应的hdfs路径 /metadata/hdfs/load2sybase，然后元数据平台会启动一个整半点扫描的定时任务，将hdfs路径下的文件同步到sybase，同时删除hdfs路径下的数据。这里提到的整半点是指每个小时的半点间隔一个小时，即1点半、2点半，而不是1点、1点半、2点。因此，sybase同步的问题可以通过检查以上两个阶段来定位。

**2、Sybase同步任务验证方式**

Sybase同步任务的验证不仅要保证Sybase任务执行成功，而且要确认数据已经同步到sybase。

**第一步**：sybase任务执行成功，检查hdfs路径下有没有相应表的数据。

hadoop fs -ls /metadata/hdfs/load2sybase

如果有表示sybase任务执行成功，没有则需要检查sybase算法配置是否正确。而且sybase同步任务执行完后会把相应表的文件删除掉，如果该目录下有长时间存在的文件，则也表明同步有问题。

**第二步**：等到任务执行完第一个整半点后，查看hdfs和sybase有没有数据，如果hdfs没有，且sybase中有数据，则表示同步成功。反之，则表示同步失败。此时，需要检查相关配置：

1. Sybase和spark字段个数、类型、顺序等是否一致。
2. Sybase summaries\_tab中是否有配置sybase表的分区。且sybase分区字段必须和表里面的字段名保持一致。

**示例**：以cdma\_netmaxc\_subject\_topn\_1x\_day\_sybase为例：

多省的情况下，cdma\_netmaxc\_subject\_topn\_1x\_day在sybase表中的省id字段是provincecode,时间字段是day，因此，我们的sybase分区必须配置成:

**partitionfield="provincecode,day" partitionfieldtype="int,date"**

1. 当然也有一些特殊字段如reportdate、timestamp、provincecode等。如果有这些，则需要去检查/home/netnumen/ems/ums-server/utils/vmax-conf下面的

dataarrivalcheck.conf文件中是否有配置满足要求的字段。

**示例：**dataarrivalcheck.conf

**PartitionRelation{**

**p\_hour = ["hour"]**

**p\_date = ["day","datestamp","endtime","timestamp","reportdate"]**

**p\_week = ["week"]**

**p\_month = ["month"]**

**p\_year = ["year"]**

**p\_provinceid = ["province","provinceid"]**

**p\_provincecode = ["provincecode"]**

**p\_imsi = ["imsi"]**

**}**

如示例所示，如果spark表中的分区是等号左边的这些字段名，那么你的sybase分区中的字段名就只能是等号右边对应的字段，同时又要满足上述2中提到的条件。如果不满足，则需要在dataarrivalcheck.conf中增加相应的字段。

**第三步**：具体的同步失败原因可以通过查看

/home/netnumen/ems/ums-server/utils/vmax-metadata-manager/log

下的debug日志，注意要取距离整半点以后最近的日志查看。如想查看1点半的sybase同步任务情况，可以去离1点半以后最近的日志。然后打开日志搜索SybaseLoader关键字，找到问题日志并分析原因。

#### 1.5、webGIS同步任务验证

**1、Webgis同步任务元数据实现机制**：Webgis同步机制和sybase类似。具体可以参见1.4节关于Sybase同步的介绍。

**2、Webgis同步任务验证方式**

**第一步**：Webgis任务执行成功，检查hdfs下有没有相应表的数据。

hadoop fs -ls /metadata/hdfs/webgispreprocess

如果有表示Webgis任务执行成功，没有则需要检查Webgis算法配置是否正确。而且Webgis同步任务执行完后会把相应表的文件删除掉，如果该目录下有长时间存在的文件，则也表明同步有问题。

**第二步**：等到任务执行完第一个整半点后，查看hdfs和/home/webgisdata/grid路径下有没有数据，如果hdfs没有，/home/webgisdata/grid下有数据，则表示同步成功。反之，则表示同步失败。此时，需要检查相关配置：

1. 保证算法配置的webgis.conf文件中的字段和表字段一致
2. 保证算法配置的webgis.conf文件增加了用于生成.csv文件的信息。
3. 保证Webgis中province\_projectdata表有数据

**示例：**cdma\_c2i\_webgis.conf

**webgisinfo{**

**username = cf**

**pagename = gis**

**}**

**cdma\_subject\_c2i\_grid\_simplify\_day{**

**webgiscolumns ="RegionID,X\_Offset,Y\_Offset,c2i"**

**webgissldfilename = cdma\_c2i\_grids.sld**

**}**

**cdma\_subject\_c2i\_grid\_simplify\_week{**

**webgiscolumns ="RegionID,X\_Offset,Y\_Offset,c2i"**

**webgissldfilename = cdma\_c2i\_grids.sld**

**}**

**cdma\_subject\_c2i\_grid\_simplify\_month{**

**webgiscolumns ="RegionID,X\_Offset,Y\_Offset,c2i"**

**webgissldfilename = cdma\_c2i\_grids.sld**

**}**

**第三步**：具体的同步失败原因可以通过查看

/home/netnumen/ems/ums-server/utils/vmax-metadata-manager/log

下的debug日志，注意要取距离整半点以后最近的日志查看。如想查看1点半的Webgis同步任务情况，可以去离1点半以后最近的日志。然后打开日志搜索GetWebGisFileFromHDFS关键字，找到问题日志并分析原因。

#### 1.6、gbase同步任务验证

**1、gbase同步任务元数据实现机制**：gbase同步机制和sybase类似。具体可以参见1.4节关于Sybase同步的介绍。

**2、gbase同步任务验证方式**

**第一步**：gbase任务执行成功，检查hdfs下有没有相应表的数据。

hadoop fs -ls /metadata/hdfs/load2gbase

如果有，表示gbase任务执行成功；没有，则需要检查gbase算法配置是否正确。而且gbase同步任务执行完后会把相应表的文件删除掉，如果该目录下有长时间存在的文件，则也表明同步有问题。

**第二步**：查看数据下载同步是否成功。

等到任务执行完第一个整半点后，查看hdfs和

/home/netnumen/ems/ums-server/utils/vmax-metadata-manager/GBaseLoaderFile路径下有没有数据，如果hdfs没有，ftp路径下有数据，则表示数据下载成功。反之，则表示同步失败。下载完成后，会同步到gbase并删除hdfs路径下相应的文件。在这个过程中，需要检查相关配置：

1. gbase和spark字段个数、类型、顺序等是否一致。
2. gbase summaries\_tab中是否有配置gbase表的分区。且gbase分区字段必须和表里面的字段名保持一致。

**示例**：以cdma\_netmaxc\_subject\_topn\_1x\_day\_gbase为例：

多省的情况下，cdma\_netmaxc\_subject\_topn\_1x\_day在gbase表中的省id字段是provincecode,时间字段是day，因此，我们的gbase分区必须配置成:

**partitionfield="provincecode,day" partitionfieldtype="int,date"**

1. 当然也有一些特殊字段如reportdate、timestamp、provincecode等。如果有这些，则需要去检查/home/netnumen/ems/ums-server/utils/vmax-conf下面的

dataarrivalcheck.conf文件中是否有配置满足要求的字段。

**示例：**dataarrivalcheck.conf

**PartitionRelation{**

**p\_hour = ["hour"]**

**p\_date = ["day","datestamp","endtime","timestamp","reportdate"]**

**p\_week = ["week"]**

**p\_month = ["month"]**

**p\_year = ["year"]**

**p\_provinceid = ["province","provinceid"]**

**p\_provincecode = ["provincecode"]**

**p\_imsi = ["imsi"]**

**}**

如示例所示，如果spark表中的分区是等号左边的这些字段名，那么你的gbase分区中的字段名就只能是等号右边对应的字段，同时又要满足上述2中提到的条件。如果不满足，则需要在dataarrivalcheck.conf中增加相应的字段。

**第三步**：具体的同步失败原因可以通过查看

/home/netnumen/ems/ums-server/utils/vmax-metadata-manager/log

下的debug日志，注意要取距离整半点以后最近的日志查看。如想查看1点半的gbase同步任务情况，可以去离1点半以后最近的日志。然后打开日志搜索GBaseLoader 关键字，找到问题日志并分析原因。

## 2、常见问题及解决办法

#### 2.1、算法替换失败，导致语法出错

1. **表名和分区都替换失败**

**具体现象**：sql算法中引入的参数值替换失败，使得解析生成的sql语句中包含原始的参数信息，最终导致算法执行时候出错。

**解决方法**：检查算法 .xml 配置文件中的输入、输出参数名和 .sql 文件中引用的名称是否一致，尤其注意对参数的引用，是不是用”$ ”符合括起来。如：

**Xml：**<inputparameter parametername="topn1XDayInput"

<partition>p\_date</partition>

</inputparameter>

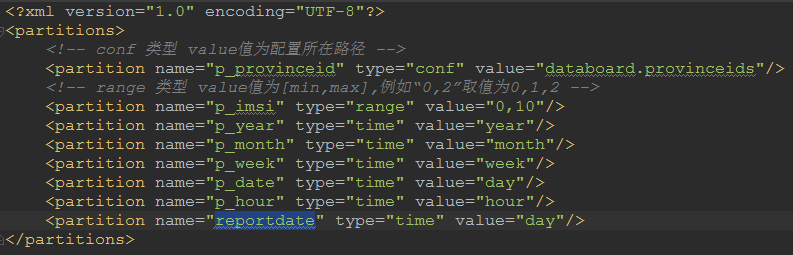
**Sql:** $topn1XDayOutput$ PARTITION(p\_date='$topn1XDayOutput.p\_date$')

1. **表名替换成功，分区替换失败**

**具体现象**：sql算法中引入的表名参数能够正常替换，但是分区信息获取失败，无法替换，最终导致算法执行的时候出错。

**解决方法**：

1. 检查上述1中提到分区信息是否配对。
2. 检查环境/home/netnumen/ems/ums-server/utils/vmax-conf目录下metadata-partitions.xml文件中是否有对相应的分区实例化。如reportdate:



1. 检查相应表的数据公告牌有没有生成。如果没有，则检查summaries\_tab\_cdma.xml中对应表的参数是否配对。之前遇到过的问题是： Location后面没加等号，导致很多任务无法替换。如：

location "/cdma/subject/cdma\_subject\_grid\_do\_bad\_week/"

#### 2.2、rdd任务问题

1. **rdd执行失败**

**具体现象**：rdd任务执行失败，报错找不到类。

**解决方法：**

1. 检查算法xml配置的架包路径中的classnames是否和代码中的匹配。

<jarpath classname="com.zte.vmax.networkoptimization.union.lteAreaEvaluateService.lteAreaEvaluateServiceRddAction" classmethod="action"> </jarpath>

1. 检查架包打的路径以及架包取的路径对不对。
2. **Rdd执行成功，但是没有数据或者分区**

**具体现象**：rdd执行成功，但是没有数据或者分区不存在，没有达到预期的效果。

**解决方法**：检查代码，增加关键位置的log打印，给主函数增加返回值。

#### 2.3、sybase同步任务问题

1. **Sybase任务显示执行成功，但是数据库没数据**

**具体现象**：sybase任务执行成功，但是sybase数据库查不到数据

**解决方法**：

1. 检查hdfs路径/metadata/hdfs/load2sybase以及

/home/netnumen/ems/ums-server/utils/vmax-metadata-manager 路径下是否有堆积的文件，如有则说明spark和sybase环境字段不匹配。

1. 检查sybase summaries\_tab中表的sybase分区是否和表字段匹配。如下所示，如果sybase中表的字段不是以下两个字段，则会导致同步之前执行删除操作失败，影响到同步。

partitionfield="provincecode,day" partitionfieldtype="int,date"

1. **Sybase任务显示成功，sybase中有部门数据**

**具体现象**：sybase同步执行成功后，sybase表中每次只有一天、一个小时或者一个省的数据，其他时间地区的数据丢失。

**解决方法**：由于sybase summaries\_tab中表的sybase分区配置不全，导致同步之前执行删除操作的时候，按值默认字段删除，导致每次都只有最后一个入库的分区数据。例如：

partitionfield=" day" partitionfieldtype=" date"

这种sybase表没配省分区信息，每次同步之前都会按照天删除历史数据，使得数据库中只会有最后一次入库的省的数据。

#### webGIS同步任务问题

**1、webGIS任务显示执行成功，但是webGIS数据库没数据**

**具体现象**：webGIS任务执行成功，但是没有数据

**解决方法**：检查sybase中是否有province\_projectdata表，如果没有手动导入数据。数据位于：

vmax-app-ran-setup\db\sybase\initdata\zh\_CN\province\_projectdata.csv

**2、webGIS任务执行失败**

**具体现象**：webGIS任务执行失败没有数据

**解决方法**：检查webgis配置文件中的字段是否和表的字段一致。

**示例**：cdma\_c2i\_webgis.conf

cdma\_subject\_c2i\_grid\_simplify\_day{

webgiscolumns ="**RegionID,X\_Offset,Y\_Offset,c2i**"

webgissldfilename = cdma\_c2i\_grids.sld

}

**3、webGIS任务执行成功，webGIS数据库有数据**

**具体现象**：webGIS任务执行成功，且数据库有数据，但是前端显示不出。

**解决方法**：检查webgis配置文件中用于生成文件名的信息是否和前端匹配。

**示例**：cdma\_c2i\_webgis.conf

webgisinfo{

username = **cf**  **----用户名username**

pagename = **gis** **----页名pagename**

}

cdma\_subject\_c2i\_grid\_simplify\_day{

webgiscolumns ="RegionID,X\_Offset,Y\_Offset,c2i"

webgissldfilename = **cdma\_c2i**\_grids.sld  **----制式,KPI**

}

最终生成的文件名为：

**cf\_gis\_cdma\_c2i**

#### gbase同步任务问题

1. **gbase任务显示执行成功，但是数据库没数据**

**具体现象**：gbase任务执行成功，但是gbase数据库查不到数据

**解决方法**：

1. 检查hdfs路径/metadata/hdfs/load2gbase以及

/home/netnumen/ems/ums-server/utils/vmax-metadata-manager/GBaseLoaderFile 路径下是否有堆积的文件，如有则说明spark到gbase同步失败。常见的问题有：

①、gbase表定义的字段类型是 datetime，但是实际的数据只有年月日，没有时分秒

②、gbase 表定义的字段个数比实际的数据中字段少，会导入失败

1. 检查gbase summaries\_tab中表的gbase分区是否和表字段匹配。如下所示，如果gbase中表的字段不是以下两个字段，则会导致同步之前执行删除操作失败，影响到同步。

partitionfield="provincecode,day" partitionfieldtype="int,date"

1. **gbase任务显示成功，gbase中有部分数据**

**具体现象**：gbase同步执行成功后，gbase表中每次只有一天、一个小时或者一个省的数据，其他时间地区的数据丢失。

**解决方法**：由于gbase summaries\_tab中表的gbase分区配置不全，导致同步之前执行删除操作的时候，按值默认字段删除，导致每次都只有最后一个入库的分区数据。例如：

partitionfield=" day" partitionfieldtype=" date"

这种gbase表没配省分区信息，每次同步之前都会按照天删除历史数据，使得数据库中只会有最后一次入库的省的数据。

#### 环境问题

1. **Nested异常**

**具体现象**：

Exception:org.apache.hadoop.hive.ql.metadata.HiveException:

checkPaths: hdfs://defaultCluster/tmp/hive-mr/hive\_2016-02-24\_00-18-50\_744\_8820725485125560857-1/-ext-10000

**has nested**

directoryhdfs://defaultCluster/tmp/hive-mr/hive\_2016-02-24\_00-18-50\_744\_8820725485125560857-1/-ext-10000/\_temporary

**解决方法**：

在任务提交集群节点的hive-site.xml中添加如下配置（vmax服务器的vmax-conf也最好加一下），然后重启集群和元数据平台。

<property>

<name>**mapred.input.dir.recursive**</name>

<value>true</value>

</property>

<property>

<name>**hive.mapred.supports.subdirectories**</name>

<value>true</value>

</property>

1. **hive metastore多版本问题异常**

**具体现象：**

Caused by: MetaException(message:Metastore contains multiple versions)

at org.apache.hadoop.hive.metastore.ObjectStore.getMSchemaVersion(ObjectStore.java:6643)

**解决方法**：需要把以下补丁放入任务提交集群的/home/mr/spark/libext路径中。

**补丁：**



1. **deleteme异常**

**具体现象**：

Caused by:

com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException: Table **'sparkomm.deleteme**1454040454438' doesn't exist

**解决方法**：

hive使用的datanucleus框架的bug。在Hive-Site.xml增加如下配置：

<property>

<name>**datanucleus.autoCreateSchema**</name>

<value>**false**</value>

</property>

<property>

<name>**datanucleus.fixedDatastore**</name>

<value>**true**</value>

</property>

## 3. Gbase定位分析方法

### 3.1 V6.15.60.03B2版本之前的定位方法

#### 3.1.1 gbase同步流程的大致说明

1. 采用quartz调度执行，调度周期可配置，默认为半点执行，如1：30、2：30等
2. 加载数据之前会先删除重复数据
3. Gbase同步成功会删除hdfs上对应数据，否则数据保留在hdfs上

#### 定位方法说明

1. 在工程目录下的gbase目录下查看debug日志，路径为 /home/netnumen/ems/ums-server/utils/vmax-metadata-manager/log/GBase/GBase.debug.log ，搜索关键字 [GbaseLoader Job]：
   1. 如果出现 load data of xxxx to gbase success ， 那么表明入库成功
   2. 如果出现 some error occur at delete history data ， 那么表明入库之前删除重复数据失败，此时在日志中向上搜带有 [GBaseLoader Detail]的日志信息，查找删除的sql，定位执行删除sql失败的问题
   3. 如果出现 load data of xxx to gbase failed ! ， 那么表明入库失败，那么向上搜索带有[GBaseLoader Detail]的日志信息
2. 对于[GbaseLoader Detail]的日志信息解析如下：
   1. 如果出现 [GBaseLoader Detail] : delete duplicate data success ， 表明入库之前，删除重复数据成功，否则可能是删除重复数据失败导致入库失败
   2. 如果出现 [GBaseLoader Detail] : connect success ! ， 表明连接load\_data.sh脚本所在服务器（一般为vmax）成功，否则排查连接失败的原因，排查项包括ip、用户名、密码等
   3. 如果出现 [GBaseLoader Detail] : download file from hdfs success ! ， 表明从hdfs上下载数据文件成功，否则排查下载文件失败的原因
   4. 如果出现 [GBaseLoader Detail] : the size of data file is 0, don't need to upload to gbase ，则说明下载的文件为0，没有必要走后面的同步流程，一般情况下会成功
   5. 如果出现 [GBaseLoader Detail] : upload file to vmax success ! ， 表明将数据上传到数据加载节点成功，否则排查上传失败的原因，排查项包括ip、用户名、密码等
   6. [GBaseLoader Detail] : the cmd use vmax is xxx ， 该日志表明执行同步脚本的cmd命令
   7. 执行同步脚本后，正常情况会生成同步日志文件和同步结果文件两个文件，然后从vmax上下载进行解析

如果出现 [GBaseLoader Detail] : parse result file success ! ， 表明成功下载了同步结果文件并解析成功，这种情况下一般任务成功，如果任务失败，可能是由于部分数据入库失败导致，需要分析gbase同步日志，详细方法请参照 [gbase同步日志分析](#_Gbase同步日志的分析)

如果出现如下异常：



* + 1. 则表明没有生成结果文件，导致下载文件失败，这种情况一般是调用load\_data.sh脚本进行入库时失败导致，需要分析gbase同步日志，详细方法请参照 [gbase同步日志分析](#_Gbase同步日志的分析)

### 3.2 V6.15.60.03B2版本及之后版本的定位方法

#### 3.2.1 此版本相对于之前版本，有如下改动

1. gbase同步不再采用quartz任务调度的形式，而采用线程不间断扫描的形式，即每间隔指定时间后扫描一次（指定时间可配置，默认为10分钟），如下：



1. gbase同步失败后，会将文件从gbase同步目录 **/metadata/hdfs/load2gbase/** 移至同步失败目录 **/metadata/hdfs/load2gbaseException/** ，因为同步失败的文件即使再次同步，也有很大可能会继续失败，但是也有可能是因为偶然环境因素导致，所以采用每扫描同步目录指定次数后（该次数可配置，默认为5），扫描一次同步失败目录。
2. 如果是因为部分数据入库失败导致的gbase同步失败，比如有100条数据，入库成功55条，失败45条，那么即使再次入库也没有意义，此时会将文件从gbase同步目录 **/metadata/hdfs/load2gbase/** 移至临时保存目录 **/metadata/hdfs/load2gbaseTmpSave** ，该文件保存下来用以检查为何会有不合法数据。这种情况一般出现在将load\_data.sh脚本中的 --max\_error\_records 的值修改为大于0的场景。

对 –max\_error\_records的说明：如果将其值修改为1000，表示一次同步最多能允许有1000条记录同步失败，如果超过1000，则同步失败，如果不超过，则同步成功，该值默认为0，即只要一条数据不合法，就会导致同步失败

1. 文件超时机制，每次扫描同步失败目录时，会在同步之前扫描同步失败目录 /metadata/hdfs/load2gbaseException/ 和临时保存目录 /metadata/hdfs/load2gbaseTmpSave，筛选出超时文件并删除，超时时间可配置（默认为10天），超时时间以文件时间为准。

#### 定位方法说明

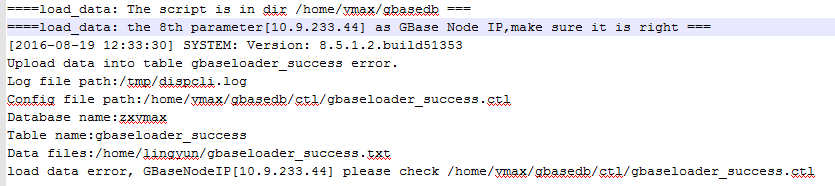
1. 在工程目录下的gbase目录下查看debug日志，路径为 /home/netnumen/ems/ums-server/utils/vmax-metadata-manager/log/GBase/ GbaseLoader.debug.log ，搜索关键字 [GbaseLoader Job]：
   1. 如果出现 load data of xxxx to gbase success ， 那么表明入库成功
   2. 如果出现 delete data of xxx to gbase failed ! ， 那么表明入库之前删除重复数据失败，此时在日志中向上搜带有 [GBaseLoader Detail]的日志信息，查找删除的sql，定位执行删除sql失败的问题
   3. 如果出现 load data of xxx to gbase half success ! ，那么表明部分数据入库失败，并且小于 max\_error\_records值（该值默认为0），此时数据文件会被移至 /metadata/hdfs/load2gbaseTmpSave/ 目录下保存，用以定位为什么存在入库失败的数据
   4. 如果出现 load data of xxx to gbase failed ! ， 那么表明入库失败，那么向上搜索带有[GBaseLoader Detail]的日志信息
2. 对于[GbaseLoader Detail]的日志信息解析如下：
   1. 如果出现 [GBaseLoader Detail] : delete duplicate data success ， 表明入库之前，删除重复数据成功，否则可能是删除重复数据失败导致入库失败
   2. 如果出现 [GBaseLoader Detail] : connect success ! ， 表明连接load\_data.sh脚本所在服务器（一般为vmax）成功，否则排查连接失败的原因，排查项包括ip、用户名、密码等
   3. 如果出现 [GBaseLoader Detail] : download file from hdfs success ! ， 表明从hdfs上下载数据文件成功，否则排查下载文件失败的原因，排查项包括ip、用户名、密码等
   4. 如果出现 [GBaseLoader Detail] : the size of data file is 0, don't need to upload to gbase ，则说明下载的文件为0，没有必要走后面的同步流程，一般情况下会成功
   5. 如果出现 [GBaseLoader Detail] : upload file to vmax success ! ， 表明将数据上传到数据加载节点成功，否则排查上传失败的原因
   6. [GBaseLoader Detail] : the cmd use vmax is xxx ， 该日志表明执行同步脚本的cmd命令
      1. [GBaseLoader Detail] : exitStatus is 0/1/2 ，表示执行cmd命令的结果，0代表成功，1代表失败，2代表入库部分失败，对于入库失败或者部分失败的情况，可以找相应的gbase同步日志定位，详细方法请参照 [gbase同步日志分析](#_Gbase同步日志的分析)

### Gbase同步日志的分析

该日志位于vmax服务器的 /home/vmax/gbasedb/GBaseLoaderFile/表名 目录下，表名为gbase同步的表名。

**常见的问题如下：**

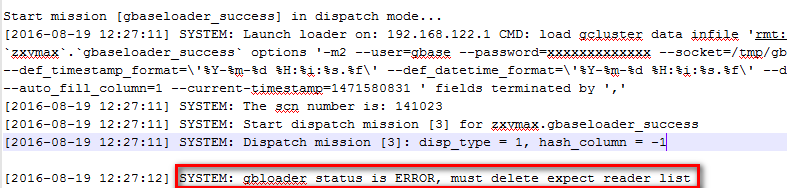
1. dispserver服务没有启动（日志如下）



如果日志中没有包含数据分发的部分，即包含 “Start mission xxx in dispatch mode...” 的信息的话，那么可能是数据加载服务没有启动

**解决办法**：在数据加载服务端启动dispserver服务即可

1. gbase数据库大小网ip配置不对（日志如下）



如果日志中出现红框标识部分的信息的话，可能是由于配置的gbase数据库ip与实际gbase安装的ip大小网不一致造成的

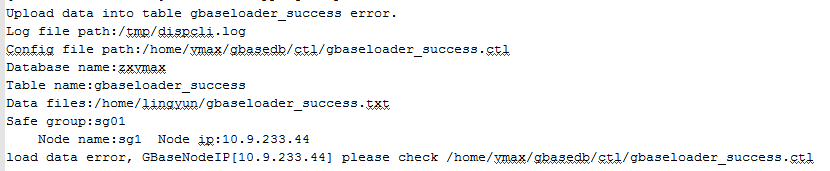
**解决办法：**修改配置serveraddress.properties中的vmaxcn.gbasehost，与gbase安装的大小网一致即可

1. 存在不合法数据（日志如下）

最常见的就是数据与表定义不一致，比如数据字段数与表定义字段数不一致、数据类型与表字段类型不一致、数据超过表字段定义的长度范围、时间类型不一致等等。失败情况大致分为下面两种：

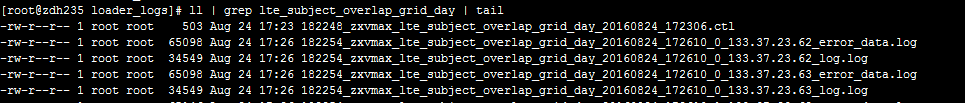
* 1. 不合法数据的记录数超过配置的max\_error\_records值（该值默认为0）

这种情况就会造成gbase同步失败，日志如下：



**解决办法：**

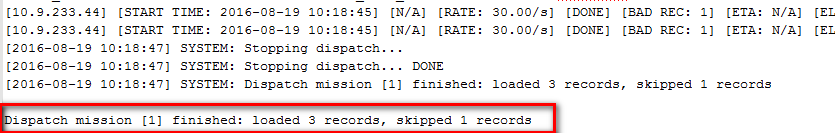
如果max\_error\_records值大于0的话，同步失败会生成日志文件，该日志存在于vmax服务器的/root/loader\_logs/目录下，如图：



通过日志可以找到同步失败的数据的位置；如果max\_error\_records值为0，那么该日志不会生成，可修改该值之后再进行定位

* 1. 不合法数据的记录数不大于配置的max\_error\_records值

这种情况下，合法的数据会入库，不合法的数据会被过滤掉，日志如下：



loaded表示入库成功的记录数，skipped表示入库失败的记录数，对于失败记录问题的定位，参照上面情况a)的解决方法

对于情况b)，在V6.15.60.03B2版本及之后版本，这种情况会视为gbase同步成功，但是数据文件会暂时保存在**/metadata/hdfs/load2gbaseTmpSave**目录下，用以定位部分数据入库失败的原因，在V6.15.60.03B2版本之前，这种情况视为入库失败

### 3.3 其他参考文档：

1. http://10.5.0.128/ZXVMAX/DOC/ZXVMAX（V6.15）/09 合作开发/GBase迁移/VMax-RAN GBase数据加载说明.docx