# 新增分区表自动安装配置程序

## 如果已有旧版分区表，请先执行已有分区表升级操作，再来执行自动安装

并且要求务必检查tlm\_tab和fragtabinfo表中是否有无关数据，如果有可能导致分片异常，从而导致数据丢失。

## 新建临时表，及其备份表

这里是针对对streamnumber不敏感的表

分片的步骤为新建要分片c2pps\_edlog的表的临时表c2pps\_edlog\_temp，之后执行分片脚本，生成正常的分片表，然后将原c2pps\_edlog表rename为c2pps\_edlog\_bak，将临时表c2pps\_edlog\_temp表rename成c2pps\_edlog，将rename后的原表c2pps\_edlog\_bak数据导出，将导出文件中的streamnumber字段置位为0，将导出的数据导入到新的分片表c2pps\_edlog中。

该过程不需要停业务，影响业务时间极短，但是会导致streamnumber混乱。如果要求streamnumber不变则需要停止业务。

根要分片的原表建立对应临时表结构，这里以mcn\_smslog 表为例

注意建立新表的时候需要遵守以下几个原则：

1. 不能有主键，所有主键都重新建立一个索引代替
2. 新建索引名不能和原表相同，统一在后缀增加\_ft
3. 所有的索引不能加in userdbs这种指定dbs的修饰

首先建立对应表的临时表，统一命名后缀增加temp

假设原表表结构如下：

create table "min01".tfcall\_calllog   
  (  
    streamnumber serial not null ,  
    callnumber char(20),  
    callid varchar(50),  
    servicekey char(8),  
    asaddress varchar(64),  
    callresult char(2),  
    callflow char(1),  
    calltime char(14),  
    primary key (streamnumber)   
  ) extent size 256000 next size 256000 lock mode row;

那么新建的temp表的表结构为

create table "min01".tfcall\_calllog\_temp  
  (  
    streamnumber serial not null ,  
    callnumber char(20),  
    callid varchar(50),  
    servicekey char(8),  
    asaddress varchar(64),  
    callresult char(2),  
    callflow char(1),  
    calltime char(14)  
  )

fragment by expression

partition p2018090100 ((calltime >= '20180901000000'

) AND (calltime < '20180910000000' ) ) in userdbs,

partition rmd remainder in userdbs

extent size 256000 next size 256000 lock mode row;

create index idx\_streamnumber\_ft on tfcall\_calllog\_temp(streamnumber) using btree;

如果需要备份，则需要建立表名加\_old的备份表，表结构如下，要求和临时表结构相同，但是序列号字段改为int，具体如下

create table "min01".tfcall\_calllog\_temp\_old  
  (  
    streamnumber int not null ,  
    callnumber char(20),  
    callid varchar(50),  
    servicekey char(8),  
    asaddress varchar(64),  
    callresult char(2),  
    callflow char(1),  
    calltime char(14)  
  )

fragment by expression

partition p2018090100 ((calltime >= '20180901000000'

) AND (calltime < '20180910000000' ) ) in userdbs,

partition rmd remainder in userdbs

extent size 256000 next size 256000 lock mode row;

create index idx\_streamnumber\_ft\_old on tfcall\_calllog\_temp\_old(streamnumber) using btree;

## 确认现网是否存在如下表

表：TLM\_table

表：tlm\_errlog

表：fragtabinfo

存储过程：add\_par\_for\_date

存储过程：add\_par\_for\_date

上述表和存储过程，如果使用的是4.0的分区系统就备份即可。如果不是请务必先执行旧表升级，否则出现任何问题自己负责。

## 一键安装部分：

将脚本和sql文件都放在/home/informix/frag下，

chmod +x \*sh

执行步骤如下：

[informix@hdr1 frag]$ sh ./go.sh

请输入要操作的库名：newmin

请输入要操作的表名：tfcall\_calllog\_temp

请输入分区列：calltime

请输入时间粒度，单位小时：168

请输入表存放的dbspace：userdbs

请输入数据保留天数：7

请输入备份表保留天数（为0则不需要备份）：180

要操作的库名为： newmin

要操作的表名： tfcall\_calllog\_temp

分区列： calltime

时间粒度： 72

表存放的dbspace： userdbs

数据保留天数： 7

请输入备份表保留天数： 180

是否确认上述信息，请确保输入正确，否则后果自负[Y/y]：y

注意，粒度尽量和保留天数匹配，为保留天数的24倍，比如保留1天，粒度就是24.

操作过程中出现如下错误没有关系，是因为每次都要建表和建立索引，在第一次已经完成了，所以后面报错，一共3个表和1个索引会有已存在报错。

310: Table (informix.tlm\_table) already exists in database.

Error in line 20

Near character position 1

310: Table (informix.tlm\_errlog) already exists in database.

Error in line 30

Near character position 1

310: Table (informix.fragtabinfo) already exists in database.

Error in line 41

Near character position 1

316: Index (tabpar) already exists in database.

Error in line 42

Near character position 59

## 随后确认新表结构：

oncheck -pt newmin:tfcall\_calllog\_temp |grep -E 'rows|partition'

## 初始化分片：

手工执行add\_par\_for\_date.sh test(库名)

执行多次

通过查看oncheck -pt newmin:tfcall\_calllog\_temp |grep -E 'rows|partition'

确定当前分片能够覆盖当前时间

手工执行del\_par\_for\_date.sh test(库名)

执行多次

通过查看oncheck -pt newmin:tfcall\_calllog\_temp |grep -E 'rows|partition'

确定将多余分片删除到只保留刚好到保留片能覆盖天数。

如果有备份，则也是用同样方法观察备份表表结构

oncheck -pt newmin:tfcall\_calllog\_temp\_old |grep -E 'rows|partition'

## rename临时表和现网表，并导数据

rename table tfcall\_calllog to tfcall\_calllog\_bak;

rename table tfcall\_calllog\_temp to tfcall\_calllog;

## 更新TLM\_table和fragtabinfo表信息：

TLM\_table表的table\_name更改成现网表名（将所有tfcall\_calllog\_temp更换为tfcall\_calllog，将所有tfcall\_calllog\_temp\_old更换为tfcall\_calllog\_old）

Fragtabinfo表的tabname更改成现网表名（将所有tfcall\_calllog\_temp更换为tfcall\_calllog，将所有tfcall\_calllog\_temp\_old更换为tfcall\_calllog\_old）

## rename互换现网表；

将mcn\_smslogtemp和mcn\_smslog表名互换

将mcn\_smslogtemp\_old重命名为mcn\_smslog\_old

## 导入数据

注意导入数据的serial类型问题。为避免出现问题，建议导入数据时，所有serial列为0

如要求序列号不能变，则使用停止业务的方式，再考虑其他方式

## 配置cron

添加两个cron，因为程序会自己判断如何增删，所以找个闲时每天定时执行即可

50 2 \* \* \* . ./.bash\_profile;sh /home/informix/frag/add\_par\_for\_date.sh test（数据库名）

35 3 \* \* \* . ./.bash\_profile;sh /home/informix/frag/del\_par\_for\_date.sh test（数据库名）

注意增删两个脚本的执行间隔，至少错开半个小时。

建议操作后观察，并且把监控工作纳入日常例检当中

## 例检

1. 定期查看/home/informix/frag /root.log日志，注意是否有error等异常字样
2. 通过oncheck监控rmd分区里面不应该有任何数据（可以使用综合监控脚本替代手工检查）；

同样，tlm\_errlog表也不应该有数据；

1. 通过如下命令观察表和索引的分区情况，索引必须要随表的分区规则而分区，否则马上联系数据库组；

dbschema -d newmin -t tfcall\_calllog -ss

oncheck -pt newmin:tfcall\_calllog|grep -E 'rows|partition'

oncheck -pt newmin:tfcall\_calllog|grep partition|wc -l

1. 通过/home/Informix/frag目录下add\_par\_for\_date.log和del\_par\_for\_date.log来查看添加、删除分区是否正常，以及执行时间；
2. 建议当晚和转天观察分区表的顺扫情况，通过配置数据库综合监控脚本监控，具体使用方法，请见《数据库综合监控脚本使用说明》
3. 如果有大量表数据变化，勿忘统计更新。