EMS速递报表并发调度系统使用说明 v0.5

|  |  |  |  |
| --- | --- | --- | --- |
| **Num** | **Author** | **Date** | **Version** |
| 1 | 星环科技(袁 波) | 2015.10.29 | 0.1 |
| 2 | 星环科技(袁 波) | 2015.11.10 | 0.5 |
| 3 | 星环科技(袁 波) | 2016.05.16 | 0.51 |

目录

[一 总览 3](#_Toc451437635)

[二 定时调度 4](#_Toc451437636)

[三 利用Hue查看报表结果 5](#_Toc451437637)

[四 报表生成流程 7](#_Toc451437638)

[4.1 重启Inceptor Server 8](#_Toc451437639)

[4.2 合并Hyperbase数据进range表 9](#_Toc451437640)

[4.3 生成日报脚本 10](#_Toc451437641)

[4.4 清理Inceptor执行环境 10](#_Toc451437642)

[4.5 执行日报脚本 10](#_Toc451437643)

[五 手动执行报表生成任务 10](#_Toc451437644)

[六 报表开发指南 12](#_Toc451437645)

[6.1时间变量约定 12](#_Toc451437646)

[七 性能分析 13](#_Toc451437647)

[附件一 生成日报sql脚本 14](#_Toc451437648)

[附件二 并发日报运行脚本 19](#_Toc451437649)

[附件三 目录清单 19](#_Toc451437650)

# 一 总览

EMS报表并发调度系统总体架构图如下，其主要包括:一、定时调度系统；二、计算资源池；三、查询终端。



首先我们介绍定时调度系统，该系统时通过crontab定时任务来实现定时功能，通过表驱动实现并发调度和依赖调度的功能。登录TD-APP-31并进入： /home/softwares/report/将看到以下内容：

|  |  |  |
| --- | --- | --- |
| **Num** | **File** | **Describe** |
| 1 | daily\_list | 日报sql配置文件 |
| 2 | daily\_list\_concurrent | 日报并发执行配置文件 |
| 3 | daily\_list\_series | 日报串行执行配置文件 |
| 4 | daily\_reports\_sql | 日报sql归档 |
| 5 | daily\_run.sh | 生成日报sql |
| 6 | daily\_run\_once.sh | 生成迭代1次的日报sql |
| 7 | balance | 日报并发提交目录 |
| 8 | clean | 环境清理工具 |
| 9 | run\_concurrence.sh | 并发日报运行脚本 |
| 10 | logs | 日志目录 |
| 11 | merge\_into | merge控制脚本 |
| 12 | monthly\_list | 月报配置文件 |
| 13 | monthly\_reports\_sql | 月报sql归档 |
| 14 | monthly\_run.sh | 生成月报sql |
| 15 | sql\_templates | 模板sql |
| 16 | sql\_templates\_200 | 模板备份 |
| 17 | Daily.log | 日报Crontab执行日志 |
| 18 | Monthly.log | 月报Crontab执行日志 |

10.1.200.32是一台全局并发控制机器，它控制着以下12台并发提交机：10.1.200.15,10.1.200.16,10.1.200.17,10.1.200.18,10.1.200.19,10.1.200.20,10.1.200.21,10.1.200.25,10.1.200.26,10.1.200.27,10.1.200.28,10.1.200.29 以及一台依赖提交机：10.1.200.30

注：这12台并发提交机的目录/home/softwares/report下都有一份完整数据镜像。

报表执行结果可以通过hue界面运行以下sql查询获取：

每日日报log查询：

select \* from log.r\_log where reportdate='20160218' order by duringtime1 desc

每日merge log查询:

select \* from log.m\_log where mergedate='20160218' order by duringtime1 desc

/home/softwares/report/balance 目录是非常重要的并发调度和均衡控制的目录，下面将分别介绍每个脚本的使用：

1. /home/softwares/report/balance/batch\_invoke\_daily\_run.sh 用于并行生成日报sql；

2. /home/softwares/report/balance/batch\_invoke\_run\_concurrent.sh 用于并行运行日报；

3. /home/softwares/report/balance/distribute.sh用于分发report文件夹至各个并发机器上；

4. /home/softwares/report/balance/restartInceptorServer.sh 用于重启inceptorServer

5. /home/softwares/report/balance/getRunningReport.sh 用于获取正在运行的脚本

6. /home/softwares/report/balance/killAllRuningReport.sh 杀死所有正在运行的脚本

7. /home/softwares/report/balance/getSQLScripts.sh 获取日报sql个数

8. /home/softwares/report/balance/getJPS.sh 获取JPS详情

/home/softwares/report/balance/killAllRuningReport.sh

分发同步，可以在任意一台非submitter的机器上做分发同步。

# 二 定时调度

[root@TD-APP-31 ~]# crontab -l

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **定时时间** | **定时任务** | **备注** |
| 1 | 0 23 \* \* \* | /home/softwares/report/balance/restartInceptorServer.sh | 清理环境 |
| 2 | 0 0 \* \* \* | /home/softwares/report/balance/batch\_invoke\_daily\_run.sh | 生成脚本 |
| 3 | 0 0 \* \* \* | /home/softwares/report/merge\_into/merge\_run.sh | 增量更新 |
| 4 | 10 0 \* \* \* | /home/softwares/report/clean/cleanDefaultDB.sh | 清空defaultDB |
| 5 | 50 0 \* \* \* | /home/softwares/report/batch\_update/run.sh | 批量更新 |
| 6 | 0 1 \* \* \* | /home/softwares/report/balance/batch\_invoke\_run\_concurrent.sh | 报表运行 |
| 7 | 0 12 \* \* \* | /home/softwares/report/compact/run\_all.sh | 大合并 |
| 8 | 0 13 \* \* \* | /home/softwares/report/custermize/run.sh | 自定义报表 |

以上8条定时任务详细说明：

1.每日夜间23点执行inceptor重启以及环境清理任务。

2.每日凌晨0点执行定时生成日报sql；

3.每日凌晨0点执行定时meger 程序，通过增量方式导入range表中；

4.每日凌晨0:10执行定时清理defaultDB的任务以及将普通码表从hyperbase中通过全量方式导入range表中；

5.每日凌晨0:50执行批量更新，用于对range分区表的增量补充；

6.每日凌晨1点

/home/softwares/report/clean/ctas\_ems\_pmart\_code.sql

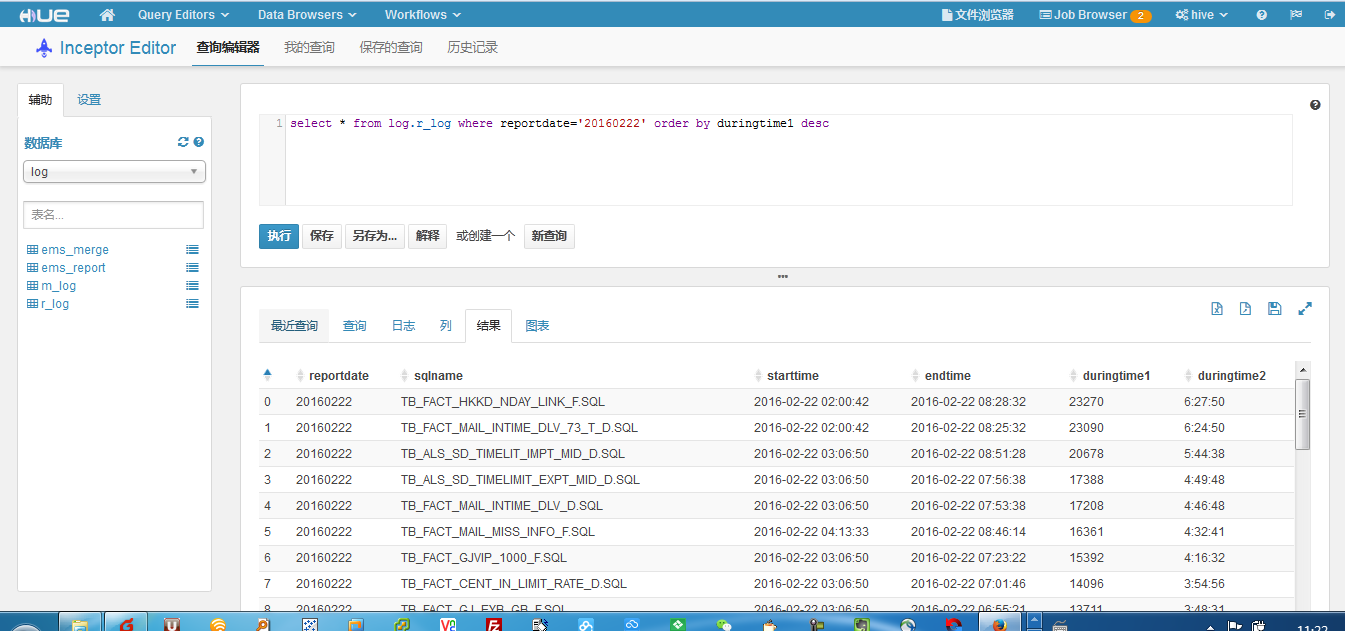
具体shell实现请参见附件一、二。日报脚本主要功能概述如下：

1. 设置Hyperbase 优化参数;
2. 删除hdfs上已存在的sql脚本;
3. 用sed替换sql模板中的如下日期：上个月最后一天，上个月第一天，上个月，去年同月，去年的上个月，昨天;
4. 根据业务意义处理后缀(\_X=1,\_K=3,\_D=7,\_F=15,\_U=30)并生成sql;
5. 上传新生成的sql文件到hdfs上.

# 三 利用Hue查看报表结果

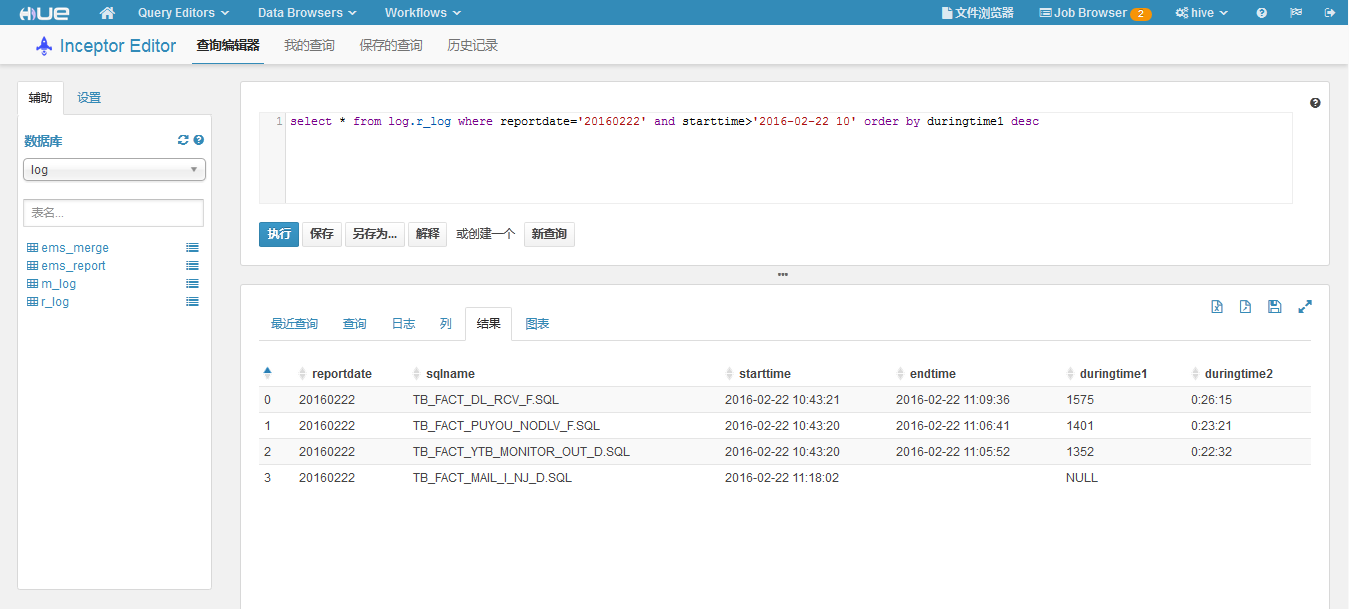
3.1查看当日报表结果：

select \* from log.r\_log where reportdate='20160222' order by duringtime1 desc



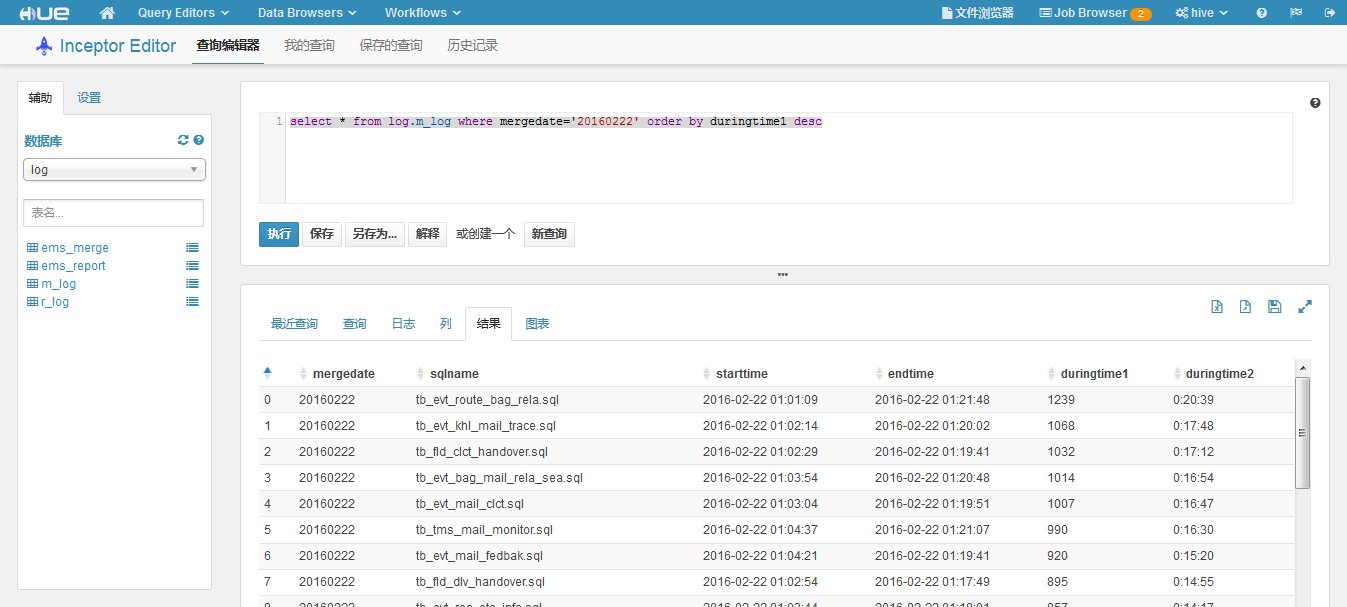
3.2 查看当日指定开始时间段的报表

select \* from log.r\_log where reportdate='20160222' and starttime>'2016-02-22 10' order by duringtime1 desc



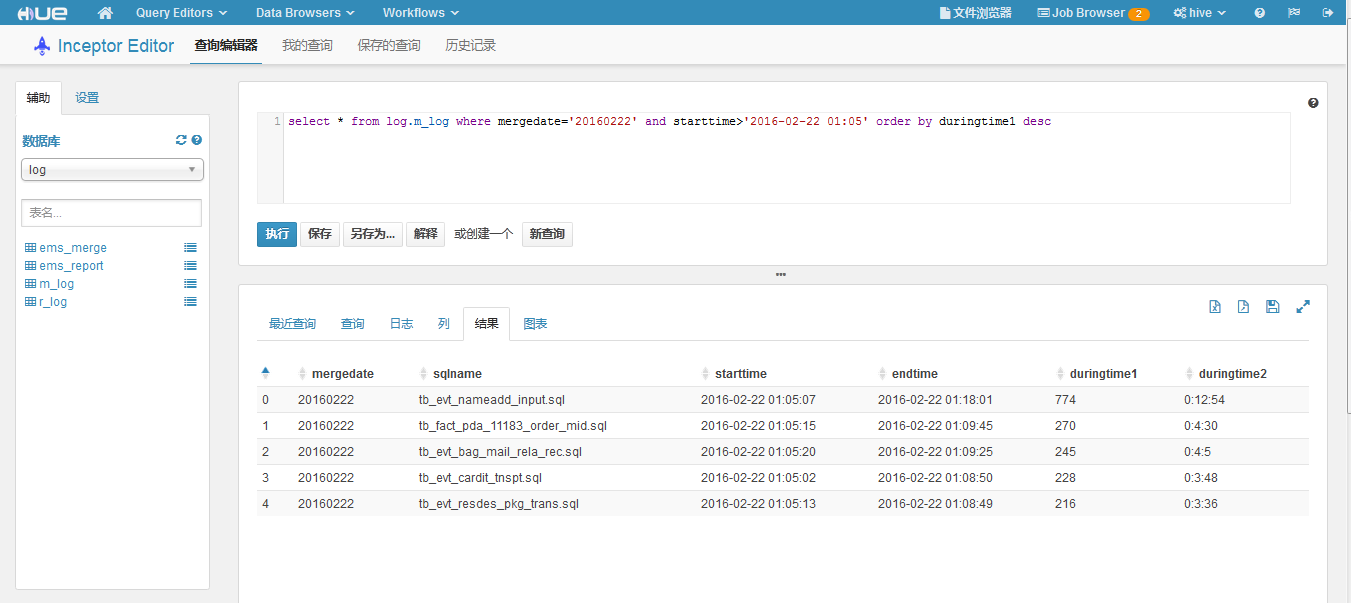
3.3查看当日merge结果

select \* from log.m\_log where mergedate='20160222' order by duringtime1 desc



3.4查看当日指定开始时间段的merge结果

select \* from log.m\_log where mergedate='20160222' and starttime>'2016-02-22 01:05' order by duringtime1 desc



# 四 报表生成流程

根据crontab定时驱动，整个报表生成步骤如下：

1、 23:00 重启Inceptor Server：防止和用于清理未结束的stage和归零stage id；

2、 01:00 合并Hyperbase数据进range表：每日增量合并；

3、 01:00 生成日报脚本：根据日报模板，生成每日日报脚本；

4、 01:10 清理Inceptor执行环境：包括情况defaultDB，全量抽码表，置位adminDB

5、 02:00 执行日报脚本：执行报表；



## 4.1 重启Inceptor Server



注意重启inceptorServer的时候是分两步，第一步是重启metastore，第二步是停止hive-server，等待300s后再启动hive-server。

## 4.2 合并Hyperbase数据进range表



该步骤包括：生成每日merge脚本和并发执行merge脚本两步操作，生成merge脚本的时间大约是2分钟左右，执行merge脚本的时间控制在1个小时以内，如果1小时结束不了需要人工调整hbase表的TTL时间（目前是7天）用于控制数据量的大小以便于并发的merge程序能在1小时内跑完。

## 4.3 生成日报脚本

## 4.4 清理Inceptor执行环境

## 4.5 执行日报脚本

# 五 手动执行报表生成任务

TD-APP-31是全局并发控制机，其也可以作为手动执行报表任务的执行机。手动执行报表生成任务的步骤如下：

1. 添加报表名称至：/home/softwares/report/daily\_list
2. 添加需要串行执行的报表名称至：/home/softwares/report/ daily\_list\_series
3. 添加需要并发执行的报表名称至：/home/softwares/report/daily\_list\_concurrent
4. 生成报表sql脚本：/home/softwares/report/ daily\_run.sh
5. 执行报表sql脚本：/home/softwares/report/ run\_concurrence.sh

# 六 报表开发指南

## 6.1时间变量约定

1.前一日减去：qyrjq1, qyrjq\_10, qyrjq\_\_100

2.前一日加上：qyrjs1, qyrjs\_10

3.上月末加上：symjs7

4.上月初减去：sycjq1, sycjq3, sycjq7, sycjq\_14

参考如下：

qyrjq1=$(date -d "$LastDay -1 days" +%Y%m%d)

qyrjq2=$(date -d "$LastDay -2 days" +%Y%m%d)

qyrjq3=$(date -d "$LastDay -3 days" +%Y%m%d)

qyrjq4=$(date -d "$LastDay -4 days" +%Y%m%d)

qyrjq5=$(date -d "$LastDay -5 days" +%Y%m%d)

qyrjq6=$(date -d "$LastDay -6 days" +%Y%m%d)

qyrjq7=$(date -d "$LastDay -7 days" +%Y%m%d)

qyrjq8=$(date -d "$LastDay -8 days" +%Y%m%d)

qyrjq9=$(date -d "$LastDay -9 days" +%Y%m%d)

qyrjq\_10=$(date -d "$LastDay -10 days" +%Y%m%d)

qyrjq\_11=$(date -d "$LastDay -11 days" +%Y%m%d)

qyrjq\_12=$(date -d "$LastDay -12 days" +%Y%m%d)

qyrjq\_13=$(date -d "$LastDay -13 days" +%Y%m%d)

qyrjq\_14=$(date -d "$LastDay -14 days" +%Y%m%d)

qyrjq\_15=$(date -d "$LastDay -15 days" +%Y%m%d)

qyrjq\_25=$(date -d "$LastDay -25 days" +%Y%m%d)

qyrjq\_29=$(date -d "$LastDay -29 days" +%Y%m%d)

qyrjq\_30=$(date -d "$LastDay -30 days" +%Y%m%d)

qyrjq\_31=$(date -d "$LastDay -31 days" +%Y%m%d)

qyrjq\_32=$(date -d "$LastDay -32 days" +%Y%m%d)

qyrjq\_45=$(date -d "$LastDay -45 days" +%Y%m%d)

qyrjq\_60=$(date -d "$LastDay -60 days" +%Y%m%d)

qyrjq\_61=$(date -d "$LastDay -61 days" +%Y%m%d)

qyrjq\_62=$(date -d "$LastDay -62 days" +%Y%m%d)

qyrjq\_90=$(date -d "$LastDay -90 days" +%Y%m%d)

qyrjq\_91=$(date -d "$LastDay -91 days" +%Y%m%d)

qyrjq\_92=$(date -d "$LastDay -92 days" +%Y%m%d)

qyrjq\_93=$(date -d "$LastDay -93 days" +%Y%m%d)

qyrjq\_99=$(date -d "$LastDay -99 days" +%Y%m%d)

qyrjq\_\_100=$(date -d "$LastDay -100 days" +%Y%m%d)

qyrjq\_\_122=$(date -d "$LastDay -122 days" +%Y%m%d)

qyrjq\_\_180=$(date -d "$LastDay -180 days" +%Y%m%d)

qyrjq\_\_186=$(date -d "$LastDay -186 days" +%Y%m%d)

qyrjq\_\_365=$(date -d "$LastDay -365 days" +%Y%m%d)

qyrjs1=$(date -d "$LastDay 1 days" +%Y%m%d)

qyrjs2=$(date -d "$LastDay 2 days" +%Y%m%d)

qyrjs3=$(date -d "$LastDay 3 days" +%Y%m%d)

qyrjs4=$(date -d "$LastDay 4 days" +%Y%m%d)

qyrjs5=$(date -d "$LastDay 5 days" +%Y%m%d)

qyrjs6=$(date -d "$LastDay 6 days" +%Y%m%d)

qyrjs7=$(date -d "$LastDay 7 days" +%Y%m%d)

qyrjs8=$(date -d "$LastDay 8 days" +%Y%m%d)

qyrjs9=$(date -d "$LastDay 9 days" +%Y%m%d)

qyrjs\_10=$(date -d "$LastDay 10 days" +%Y%m%d)

qyrjs\_11=$(date -d "$LastDay 11 days" +%Y%m%d)

qyrjs\_12=$(date -d "$LastDay 12 days" +%Y%m%d)

qyrjs\_13=$(date -d "$LastDay 13 days" +%Y%m%d)

qyrjs\_14=$(date -d "$LastDay 14 days" +%Y%m%d)

qyrjs\_15=$(date -d "$LastDay 15 days" +%Y%m%d)

qyrjs\_20=$(date -d "$LastDay 20 days" +%Y%m%d)

qyrjs\_21=$(date -d "$LastDay 21 days" +%Y%m%d)

qyrjs\_30=$(date -d "$LastDay 30 days" +%Y%m%d)

qyrjs\_31=$(date -d "$LastDay 31 days" +%Y%m%d)

qyrjs\_33=$(date -d "$LastDay 33 days" +%Y%m%d)

symjs7=$(date -d "$LastMonthL 7 days" +%Y%m%d)

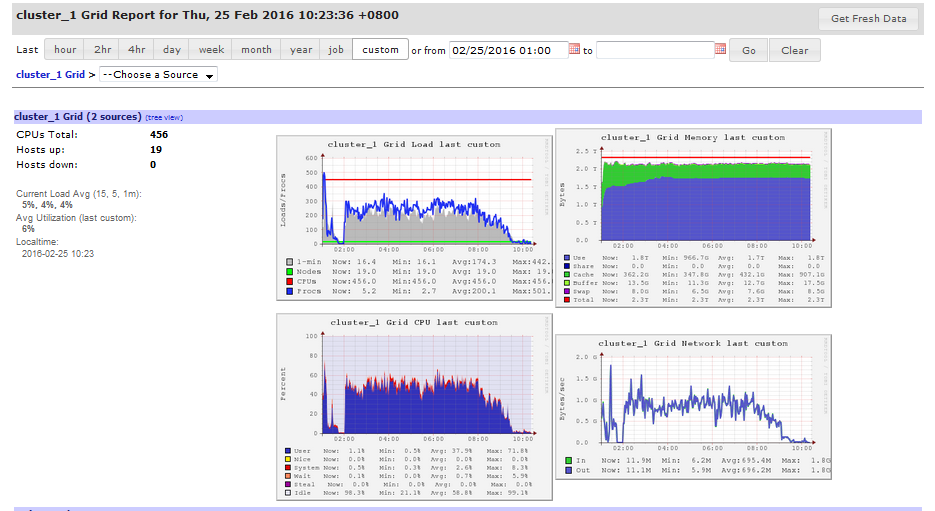
sycjq1=$(date -d "$LastMonthF -1 days" +%Y%m%d)

sycjq3=$(date -d "$LastMonthF -3 days" +%Y%m%d)

sycjq7=$(date -d "$LastMonthF -7 days" +%Y%m%d)

sycjq\_14=$(date -d "$LastMonthF -14 days" +%Y%m%d)

# 七 性能分析

****

# 附件一 生成日报sql脚本

#!/bin/sh

#transwarp="beeline -u 'jdbc:hive2://TD-APP-23:10000/default'"

prefix\_001="set hyperbase.reader=true;"

prefix\_002="set hyperbase.reader.split.size=1024000000;"

prefix\_003="set hyperbase.auto.flush=true;"

prefix\_004="set ngmr.o4.join=true;"

#prefix\_0041="set inceptor.optimizer.on=true;"

#prefix\_0042="set ngmr.reader.rowfilter=false;"

prefix\_005="set orc.crud.optimized=false;"

prefix\_006="set mapred.reduce.tasks=24;"

base\_home="/home/softwares/report/"

template\_dir=${base\_home}"sql\_templates/"

today\_tmp=`date +%Y%m%d`

#today\_tmp=$(date -d "20151225" +%Y%m%d);

daily\_dir=${base\_home}"daily\_reports\_sql/"${today\_tmp}"/"

rm -rf "$daily\_dir"

mkdir "$daily\_dir"

#sudo -u hive hadoop fs -rm -r /user/hive/daily\_reports\_sql

#sudo -u hive hadoop fs -mkdir /user/hive/daily\_reports\_sql

for target in `cat ${base\_home}daily\_list`;

do

prefix\_start="insert into log.ems\_report values('${today\_tmp}\_${target}','${today\_tmp}','${target}',sysdate,'');"

prefix\_end="update log.ems\_report set endtime=sysdate where key='${today\_tmp}\_${target}';"

tmp\_sql\_path=${template\_dir}${target}

daily\_sql\_path=${daily\_dir}${target}

#echo $prefix\_001 >> $daily\_sql\_path

#echo $prefix\_002 >> $daily\_sql\_path

echo $prefix\_004 >> $daily\_sql\_path

#echo $prefix\_0041 >> $daily\_sql\_path

#echo $prefix\_0042 >> $daily\_sql\_path

echo $prefix\_005 >> $daily\_sql\_path

#echo $prefix\_006 >> $daily\_sql\_path

echo $prefix\_start >> $daily\_sql\_path

count=1

if [[ $target =~ .\*\_X\. ]]

then

count=1

fi

if [[ $target =~ .\*\_K\. ]]

then

count=3

fi

if [[ $target =~ .\*\_D\. ]]

then

count=7

fi

if [[ $target =~ .\*\_F\. ]]

then

count=15

fi

if [[ $target =~ .\*\_U\. ]]

then

count=30

fi

# count=1

if [[ $count -ne 0 ]]

then

for (( ii=0; ii<$count; ii++ ))

do

USER\_DATE=$(date -d "$today\_tmp -$ii days" +%Y%m%d);

#echo "--------------------USER\_DATE:${USER\_DATE}"

FIRST\_DAY\_CURR\_MONTH=$(date -d "$USER\_DATE" +%Y%m01)

LastMonthL=$(date -d "$FIRST\_DAY\_CURR\_MONTH -1 day" +%Y%m%d)

LastMonthF=$(date -d "$USER\_DATE -1 month" +%Y%m01)

LastLastMonth=$(date -d "$USER\_DATE -2 month" +"%Y%m")

LastMonth=$(date -d "$USER\_DATE -1 month" +"%Y%m")

LastYearLastMonth=$(date -d "$USER\_DATE 1 year ago 1 month ago" +"%Y%m")

LastDay=$(date -d "$USER\_DATE -1 days" +%Y%m%d)

qyrjq1=$(date -d "$LastDay -1 days" +%Y%m%d)

qyrjq2=$(date -d "$LastDay -2 days" +%Y%m%d)

qyrjq3=$(date -d "$LastDay -3 days" +%Y%m%d)

qyrjq4=$(date -d "$LastDay -4 days" +%Y%m%d)

qyrjq5=$(date -d "$LastDay -5 days" +%Y%m%d)

qyrjq6=$(date -d "$LastDay -6 days" +%Y%m%d)

qyrjq7=$(date -d "$LastDay -7 days" +%Y%m%d)

qyrjq8=$(date -d "$LastDay -8 days" +%Y%m%d)

qyrjq9=$(date -d "$LastDay -9 days" +%Y%m%d)

qyrjq\_10=$(date -d "$LastDay -10 days" +%Y%m%d)

qyrjq\_11=$(date -d "$LastDay -11 days" +%Y%m%d)

qyrjq\_12=$(date -d "$LastDay -12 days" +%Y%m%d)

qyrjq\_13=$(date -d "$LastDay -13 days" +%Y%m%d)

qyrjq\_14=$(date -d "$LastDay -14 days" +%Y%m%d)

qyrjq\_15=$(date -d "$LastDay -15 days" +%Y%m%d)

qyrjq\_25=$(date -d "$LastDay -25 days" +%Y%m%d)

qyrjq\_30=$(date -d "$LastDay -30 days" +%Y%m%d)

qyrjq\_31=$(date -d "$LastDay -31 days" +%Y%m%d)

qyrjq\_32=$(date -d "$LastDay -32 days" +%Y%m%d)

qyrjq\_60=$(date -d "$LastDay -60 days" +%Y%m%d)

qyrjq\_61=$(date -d "$LastDay -61 days" +%Y%m%d)

qyrjq\_62=$(date -d "$LastDay -62 days" +%Y%m%d)

qyrjq\_90=$(date -d "$LastDay -90 days" +%Y%m%d)

qyrjq\_91=$(date -d "$LastDay -91 days" +%Y%m%d)

qyrjq\_\_180=$(date -d "$LastDay -180 days" +%Y%m%d)

qyrjq\_\_365=$(date -d "$LastDay -365 days" +%Y%m%d)

qyrjs1=$(date -d "$LastDay 1 days" +%Y%m%d)

qyrjs2=$(date -d "$LastDay 2 days" +%Y%m%d)

qyrjs3=$(date -d "$LastDay 3 days" +%Y%m%d)

qyrjs4=$(date -d "$LastDay 4 days" +%Y%m%d)

qyrjs6=$(date -d "$LastDay 6 days" +%Y%m%d)

qyrjs7=$(date -d "$LastDay 7 days" +%Y%m%d)

qyrjs8=$(date -d "$LastDay 8 days" +%Y%m%d)

qyrjs9=$(date -d "$LastDay 9 days" +%Y%m%d)

qyrjs\_10=$(date -d "$LastDay 10 days" +%Y%m%d)

qyrjs\_11=$(date -d "$LastDay 11 days" +%Y%m%d)

qyrjs\_12=$(date -d "$LastDay 12 days" +%Y%m%d)

qyrjs\_13=$(date -d "$LastDay 13 days" +%Y%m%d)

qyrjs\_14=$(date -d "$LastDay 14 days" +%Y%m%d)

qyrjs\_15=$(date -d "$LastDay 15 days" +%Y%m%d)

symjs7=$(date -d "$LastMonthL 7 days" +%Y%m%d)

sycjq1=$(date -d "$LastMonthF -1 days" +%Y%m%d)

sycjq3=$(date -d "$LastMonthF -3 days" +%Y%m%d)

sycjq7=$(date -d "$LastMonthF -7 days" +%Y%m%d)

sycjq\_14=$(date -d "$LastMonthF -14 days" +%Y%m%d)

#echo "lastday::${LastDay}"

#tmp\_sql\_path

sed "s,ssyymm,$LastMonthL,g" $tmp\_sql\_path >> $daily\_sql\_path

sed -i "s,ssyycc,$LastMonthF,g" $daily\_sql\_path

sed -i "s,ssssyy,$LastLastMonth,g" $daily\_sql\_path

sed -i "s,ssyyyy,$LastMonth,g" $daily\_sql\_path

sed -i "s,ssnnmm,$LastYearLastMonth,g" $daily\_sql\_path

sed -i "s,qqyyrr,$LastDay,g" $daily\_sql\_path

sed -i "s,qyrjq1,$qyrjq1,g" $daily\_sql\_path

sed -i "s,qyrjq2,$qyrjq2,g" $daily\_sql\_path

sed -i "s,qyrjq3,$qyrjq3,g" $daily\_sql\_path

sed -i "s,qyrjq4,$qyrjq4,g" $daily\_sql\_path

sed -i "s,qyrjq5,$qyrjq5,g" $daily\_sql\_path

sed -i "s,qyrjq6,$qyrjq6,g" $daily\_sql\_path

sed -i "s,qyrjq7,$qyrjq7,g" $daily\_sql\_path

sed -i "s,qyrjq8,$qyrjq8,g" $daily\_sql\_path

sed -i "s,qyrjq9,$qyrjq9,g" $daily\_sql\_path

sed -i "s,qyrjq\_10,$qyrjq\_10,g" $daily\_sql\_path

sed -i "s,qyrjq\_11,$qyrjq\_11,g" $daily\_sql\_path

sed -i "s,qyrjq\_12,$qyrjq\_12,g" $daily\_sql\_path

sed -i "s,qyrjq\_13,$qyrjq\_13,g" $daily\_sql\_path

sed -i "s,qyrjq\_14,$qyrjq\_14,g" $daily\_sql\_path

sed -i "s,qyrjq\_15,$qyrjq\_15,g" $daily\_sql\_path

sed -i "s,qyrjq\_25,$qyrjq\_25,g" $daily\_sql\_path

sed -i "s,qyrjq\_25,$qyrjq\_25,g" $daily\_sql\_path

sed -i "s,qyrjq\_30,$qyrjq\_30,g" $daily\_sql\_path

sed -i "s,qyrjq\_32,$qyrjq\_32,g" $daily\_sql\_path

sed -i "s,qyrjq\_31,$qyrjq\_31,g" $daily\_sql\_path

sed -i "s,qyrjq\_60,$qyrjq\_60,g" $daily\_sql\_path

sed -i "s,qyrjq\_61,$qyrjq\_61,g" $daily\_sql\_path

sed -i "s,qyrjq\_62,$qyrjq\_62,g" $daily\_sql\_path

sed -i "s,qyrjq\_90,$qyrjq\_90,g" $daily\_sql\_path

sed -i "s,qyrjq\_91,$qyrjq\_91,g" $daily\_sql\_path

sed -i "s,qyrjq\_\_180,$qyrjq\_\_180,g" $daily\_sql\_path

sed -i "s,qyrjq\_\_365,$qyrjq\_\_365,g" $daily\_sql\_path

sed -i "s,qyrjs1,$qyrjs1,g" $daily\_sql\_path

sed -i "s,qyrjs2,$qyrjs2,g" $daily\_sql\_path

sed -i "s,qyrjs3,$qyrjs3,g" $daily\_sql\_path

sed -i "s,qyrjs4,$qyrjs4,g" $daily\_sql\_path

sed -i "s,qyrjs6,$qyrjs6,g" $daily\_sql\_path

sed -i "s,qyrjs7,$qyrjs7,g" $daily\_sql\_path

sed -i "s,qyrjs8,$qyrjs8,g" $daily\_sql\_path

sed -i "s,qyrjs9,$qyrjs4,g" $daily\_sql\_path

sed -i "s,qyrjs\_10,$qyrjs\_10,g" $daily\_sql\_path

sed -i "s,qyrjs\_11,$qyrjs\_11,g" $daily\_sql\_path

sed -i "s,qyrjs\_12,$qyrjs\_12,g" $daily\_sql\_path

sed -i "s,qyrjs\_13,$qyrjs\_13,g" $daily\_sql\_path

sed -i "s,qyrjs\_14,$qyrjs\_14,g" $daily\_sql\_path

sed -i "s,qyrjs\_15,$qyrjs\_15,g" $daily\_sql\_path

sed -i "s,symjs7,$symjs7,g" $daily\_sql\_path

sed -i "s,sycjq1,$sycjq1,g" $daily\_sql\_path

sed -i "s,sycjq3,$sycjq3,g" $daily\_sql\_path

sed -i "s,sycjq7,$sycjq7,g" $daily\_sql\_path

sed -i "s,sycjq\_14,$sycjq\_14,g" $daily\_sql\_path

echo " " >> $daily\_sql\_path

echo "Daily Info: Date:`date` Lastday:${LastDay} Template:${target}">>${base\_home}logs/daily/${today\_tmp}.log

done

fi

sed -i "s,ems\_pdata\.,ems\_pdata\_range\.,g" $daily\_sql\_path

sed -i "s,EMS\_PDATA\.,ems\_pdata\_range\.,g" $daily\_sql\_path

#sed -i "s,ems\_pmart\.,ems\_pmart\_test\.,g" $daily\_sql\_path

#sed -i "s,EMS\_PMART\.,ems\_pmart\_test\.,g" $daily\_sql\_path

echo " " >> $daily\_sql\_path

echo $prefix\_end >> $daily\_sql\_path

done

#sudo -u hive hadoop fs -put ${daily\_dir}\*.\* /user/hive/daily\_reports\_sql/

# 附件二 并发日报运行脚本

#!/bin/sh

today\_tmp=`date +%Y%m%d`

#today\_tmp=$(date -d "20151225" +%Y%m%d);

prefix='/home/softwares/report/daily\_reports\_sql/'${today\_tmp}'/'

rm -rf /home/softwares/report/logs/report/${today\_tmp}

mkdir /home/softwares/report/logs/report/${today\_tmp}

for target in `cat /home/softwares/report/daily\_list\_series`;

do

location=${prefix}${target}

nohup beeline -u 'jdbc:hive2://10.1.200.24:10000/default' -f $location >>/home/softwares/report/logs/report/${today\_tmp}/$target

done

for target in `cat /home/softwares/report/daily\_list\_concurrent`;

do

location=${prefix}${target}

nohup beeline -u 'jdbc:hive2://10.1.200.24:10000/default' -f $location >>/home/softwares/report/logs/report/${today\_tmp}/$target &

done

# 附件三 目录清单

./report

├── balance

│   ├── batch\_invoke\_daily\_run.log

│   ├── batch\_invoke\_daily\_run.sh

│   ├── batch\_invoke.log

│   ├── batch\_invoke\_run\_concurrent.log

│   ├── batch\_invoke\_run\_concurrent.sh

│   ├── batch\_invoke\_run\_concurrent.sh\_bak

│   ├── batch\_invoke.sh

│   ├── concurrent\_server.tmp

│   ├── depend\_complete\_list.tmp

│   ├── depend\_mid.sh

│   ├── depend\_runable\_list.tmp

│   ├── distribute.sh

│   ├── generate\_depend\_run\_concurrent.sh

│   ├── generate\_run\_concurrent.sh

│   ├── getFileCount.sh

│   ├── getFree.sh

│   ├── getJPS.sh

│   ├── getLog.sh

│   ├── getRunningReport.sh

│   ├── getSQLScripts.sh

│   ├── invoke.sh

│   ├── iplists

│   ├── killAllRuningReport.sh

│   ├── log

│   │   ├── report\_log

│   │   └── report\_sql

│   ├── manual\_batch\_invoke\_run\_concurrent.sh

│   ├── nohup.out

│   ├── restartInceptorServer.log

│   ├── restartInceptorServer.sh

│   ├── scp

│   │   ├── concurrent\_all\_report.tmp

│   │   ├── concurrent\_daily\_list\_1

│   │   ├── concurrent\_daily\_list\_2

│   │   ├── concurrent\_daily\_list\_3

│   │   ├── concurrent\_daily\_list\_4

│   │   ├── concurrent\_daily\_list\_5

│   │   ├── concurrent\_daily\_list\_concurrent\_1

│   │   ├── concurrent\_daily\_list\_concurrent\_2

│   │   ├── concurrent\_daily\_list\_concurrent\_3

│   │   ├── concurrent\_daily\_list\_concurrent\_4

│   │   ├── concurrent\_daily\_list\_concurrent\_5

│   │   ├── concurrent\_daily\_list\_series\_1

│   │   ├── concurrent\_daily\_list\_series\_2

│   │   ├── concurrent\_daily\_list\_series\_3

│   │   ├── concurrent\_daily\_list\_series\_4

│   │   ├── concurrent\_daily\_list\_series\_5

│   │   ├── concurrent\_server.tmp

│   │   ├── depend\_all\_report.tmp

│   │   ├── depend\_daily\_list\_1

│   │   ├── depend\_daily\_list\_concurrent\_1

│   │   ├── depend\_daily\_list\_series\_1

│   │   ├── depend\_server.tmp

│   │   ├── get\_concurrent\_dailylist.sh

│   │   ├── get\_depend\_dailylist.sh

│   │   ├── scp\_concurrent.sh

│   │   └── scp\_depend.sh

├── check

│   ├── check.sh

│   ├── daily\_list\_1

│   ├── daily\_list\_2

│   ├── daily\_list\_3

│   ├── daily\_list\_4

│   ├── daily\_list\_5

│   ├── daily\_list\_concurrent\_1

│   ├── daily\_list\_concurrent\_2

│   ├── daily\_list\_concurrent\_3

│   ├── daily\_list\_concurrent\_4

│   ├── daily\_list\_concurrent\_5

│   ├── getdailylist.sh

│   └── result.txt

├── clean

│   ├── cleanDefaultDB.log

│   ├── cleanDefaultDB.sh

│   ├── cleannoctas.sh

│   ├── clean.sh

│   ├── ctas\_ems\_pmart\_code.sql

│   ├── logs

│   ├── nohup.out

│   └── temptables.txt

├── daily\_list

├── daily\_list\_concurrent

├── daily\_list\_series

├── daily.log

├── daily\_reports\_sql

├── merge\_into

│   ├── daily\_list

│   ├── daily\_list.bak

│   ├── daily.log

│   ├── daily\_merge\_sql

│   ├── daily\_run.sh

│   ├── daily\_run.sh\_bak

│   ├── logs

│   │   ├── daily

│   │   └── merge

│   ├── manual\_merge.sh

│   ├── manual\_specify\_date\_merge.sh

│   ├── merge.log

│   ├── merge\_run.sh

│   └── sql\_templates

│   ├── tb\_api\_orders.sql

│   ├── tb\_eub\_orders.sql

│   ├── tb\_evt\_airline\_info.sql

│   └── .............................

├── month.log

├── monthly\_list

├── monthly\_reports\_sql

├── monthly\_run.sh

├── nohup.out

├── report\_list

├── run\_concurrence.log

├── run\_concurrence.sh

├── run\_reports.sh

├── sql\_templates

│   ├── TB\_ALS\_SD\_TIMELIMIT\_EXPT\_MID\_D.SQL

│   ├── TB\_ALS\_SD\_TIMELIT\_IMPT\_MID\_D.SQL

│   ├── TB\_EVT\_BAG\_MAIL\_RELA\_SEA\_MID\_X.SQL

│   ├── TB\_EVT\_YH\_JS\_D.SQL

│ ├── ...................................