易数报表开发指南

**1. 总体逻辑**



**2. 数据统计指标定义文件**

数据统计指标定义文件，该文件默认存放在报表统计模块的$REPORT\_HOME/config/report/index目录下。

文件样例如下，一个文件中可以包含多个统计指标的定义。

indexKey唯一标识一个统计指标，必须全局唯一，命名规则为：产业名称.统计类型.字段，如果存在同名，则后加载的指标会覆盖先前加载。

fieldName为指标在在HBase中字段名称，可以重复。

hql为统计时执行的查询语句，其中可以包含如下内置宏变量（宏变量是可选，将运行过程中被替换）：

{STARTTIME} ：本次统计的开始时间，对于于 上一次统计的时间，日表格式为yyyyMMdd，月表格式为：yyyyMM

{ENDTIME} ：本次统计的结束时间，默认为当天，日表格式为yyyyMMdd，月表格式为：yyyyMM

统计的时间范围为({STARTTIME}, {ENDTIME}) , 双开区间。

一般外层为分组和聚合函数，里层为查询语句（非强制），可以包含多个统计字段，统计字段之间以逗号分隔。

<reportIndexGroup>

<multiHqlReportIndexs>

        <multiHqlReportIndex>

<indexKey>vip.android.newvips.d</indexKey>

<fieldName>newvips</fieldName>

<hql>

<![CDATA[

select projectid, relchannel,producttype,dtime,newvips

from

(

select projectid, 'ALL' relchannel,'0' producttype,dtime, count(distinct(userid)) newvips

from

(

select projectid ,dtime,userid

from game.T\_MID\_VIPSTATUS\_ADORKJ t

where t.chargetype = 2

and t.chargeresult = 0

and t.producttype = '0'

and from\_unixtime(unix\_timestamp(substr(timestamp,1,10),'yyyy-MM-dd'),'yyyyMMdd') = dtime

) t

group by projectid, dtime

union all

select projectid, relchannel,'0' producttype,dtime, count(distinct(userid)) newvips

from

(

select projectid ,dtime,userid, relchannel

from game.T\_MID\_VIPSTATUS\_ADORKJ t

where t.chargetype = 2

and t.chargeresult = 0

and t.producttype = '0'

and from\_unixtime(unix\_timestamp(substr(timestamp,1,10),'yyyy-MM-dd'),'yyyyMMdd') = dtime

) t

group by projectid, dtime,relchannel

) tt

]]>

</hql>

</multiHqlReportIndex>

         </multiHqlReportIndexs>

</reportIndexGroup>

数据统计指标定义文件中可以使用自定义的宏变量来减少重复代码，比如：将字段的别名或过滤规则，在HQL中使用{宏变量名}，如下所示：

<multiHqlReportIndex>

<indexKey>game.showvip.jallrevenue.d</indexKey>

<macroList>

<macro>

<name>GAMEID</name>

<value>

<![CDATA[

case

when merchantid ='1000004318' then '20160216002'

when merchantid ='1000005537' then '20160411001'

when substr(merchantid,0,6) = '612001' then substr(merchantid,7)

else substr(merchantid,8)

end as gameid

]]>

</value>

</macro>

<macro>

<name>WHERE\_CLAUST</name>

<value>

<![CDATA[

where t1.date > '{STARTTIME}' AND t1.date < '{ENDTIME}'

and t1.chargeresult='0'

and t1.producttype='2'

and t1.subtype='1'

and t1.id in (1001205)

]]>

</value>

</macro>

</macroList>

<fieldName>jallrevenue</fieldName>

<hql>

<![CDATA[

select projectid,dtime,relchannel,gameid,

sum(totalfee) jallrevenue

from

(

select t1.projectid,

{GAMEID},

t1.totalfee,

from\_unixtime(unix\_timestamp(substr(t1.recordtime, 0, 10),'yyyy-MM-dd'),'yyyyMMdd') dtime,

if(relchannel is null,'none',if(trim(relchannel) = '', 'none', if(substr(relchannel,0,1) = 'l', substr(relchannel,4), relchannel))) as relchannel

from game.t\_cdr\_payment t1

{WHERE\_CLAUST}

union all

select t1.projectid,

{GAMEID},

t1.totalfee,

from\_unixtime(unix\_timestamp(substr(t1.recordtime, 0, 10),'yyyy-MM-dd'),'yyyyMMdd') dtime,

'ALL' relchannel

from game.t\_cdr\_payment t1

{WHERE\_CLAUST}

union all

select t1.projectid,

'ALL' gameid,

t1.totalfee,

from\_unixtime(unix\_timestamp(substr(t1.recordtime, 0, 10),'yyyy-MM-dd'),'yyyyMMdd') dtime,

'ALL' relchannel

from game.t\_cdr\_payment t1

{WHERE\_CLAUST}

) t

group by projectid,dtime,gameid,relchannel

]]>

</hql>

</multiHqlReportIndex>

**3. 报表定义文件**

报表定义文件，该文件默认存放在报表统计模块的$REPORT\_HOME/config/report目录下

一个文件中只能有一个reportSuite和一个reports，一个reports可以有多个report，一个report对应一个报表，可以包含多个field，多个dimension（维度），一个dimension可以包含多个reportIndex。

report： tableName 对应于HBase中的表名。keyFields为统计表中主键字段。Period为统计周期，取值为：daily/weekly/monthly，timeFormat为时间字段的格式，timeField为时间字段的字段名

fieldDefinition为报表主键的定义，可以包含多个field，field定义如下：name和keyFields对应，regular为关键字段正则表达式，formula为显示时转换公式。

dimension：id 为报表中维度的唯一标识，可以是字母或数字，不重复即可。逻辑上有依赖关系的指标或需要独占计算资源的指标单独定义dimension。

reportIndex： 维度中包含的统计指标，在指标定义文件中indexKey中一致，逻辑上互相独立的指标放在同一个reportIndex中。

可以将统计中常用的子查询定义为物化视图，以减少数据的计算量。

<reportSuite>

         <reports>

                   <report tableName = "t\_partner\_summary\_d" keyFields = "projectid,partnerid,dtime"

                                     period = "daily" timeFormat = "yyyyMMdd" timeField = "dtime">

                            <fieldDefinition>

                                     <field name="projectid" regular="[0-9a-zA-Z./-]\*"/>

                                     <field name="partnerid" regular="[0-9a-zA-Z./-]\*"/>

                                     <field name="dtime" formula="dtime" regular="[0-9a-zA-Z./-]\*"/>

                            </fieldDefinition>

                           <materialViewers>

<materialViewer tableName = "game.T\_MID\_VIPSTATUS" clear="false">

<hql>

<![CDATA[

SELECT userid, projectid, timestamp,chargetype,chargeresult,relchannel,fee,dtime,unsubchannelid from

(

select \*,row\_number() over (partition by userid,projectid,dtime,relchannel,chargetype,timestamp,chargeresult,fee) AS rn

from

(

SELECT userid, projectid, timestamp,chargetype,chargeresult,relchannel,fee,dtime,unsubchannelid

from

(

SELECT t2.userid, t2.projectid,t1.dtime, t2.timestamp,t2.chargetype,t2.chargeresult,t2.fee,t2.relchannel,t2.unsubchannelid,

RANK() OVER (partition by t2.userid,t2.projectid,t1.dtime ORDER BY timestamp DESC,chargetype DESC) AS rank

from

(

select from\_unixtime(unix\_timestamp(dtime,'yyyy-MM-dd'),'yyyyMMdd') dtime

from game.t\_bdi\_rundate

where from\_unixtime(unix\_timestamp(dtime,'yyyy-MM-dd'),'yyyyMMdd') > '{STARTTIME}'

AND from\_unixtime(unix\_timestamp(dtime,'yyyy-MM-dd'),'yyyyMMdd') < '{ENDTIME}'

) t1,

(

SELECT original\_msisdn userid, oaoperatorid projectid,timestamp,chargetype,chargeresult,fee,

if(vipsubchannelid is null, 'none', if(substr(vipsubchannelid, 0, 1) = 'l', substr(vipsubchannelid, 4), if(accesschannel = 214, concat\_ws('%',CONV( accesschannel,10,10),vipsubchannelid), vipsubchannelid))) relchannel

, date dtime,unsubchannelid

FROM game.t\_cdr\_order

where (producttype <> '2' or producttype is null) and date > '{STARTTIME}' and date < '{ENDTIME}'

union all

SELECT userid, projectid, timestamp,chargetype,chargeresult,fee,relchannel,dtime,unsubchannelid

from game.T\_MID\_VIPSTATUS

) t2

where t2.dtime <= t1.dtime

) t

where t.rank = 1

) t

) t

where rn = 1

]]>

</hql>

</materialViewer>

</materialViewers>

                            <dimensions>

                                     <dimension id = "0">

                                               <reportIndex>

                                                        game.partner.accusercnt,

                                                        game.partner.accpayamount

                                               </reportIndex>

                                     </dimension>

                            </dimensions>

                   </report>

         </reports>

</reportSuite>

**4. 使用方法：**

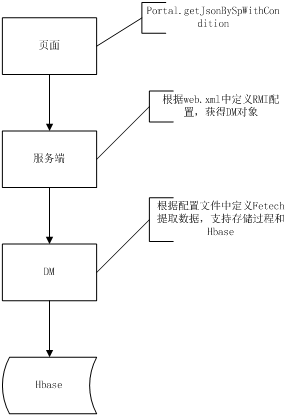
1. 开发好index文件和report文件后，将index文件上传到统计模块的默认目录下。
2. 在统计模块的bin目录下，执行./start.sh -t 报表文件名（带路径），如下所示，-t 为报表文件名，-s 为统计的开始时间，-e 为统计的结束时间，-u为读取HIVE库和写HBase表的Hadoop用户 –k 为读取HIVE库和写HBase表的Kerbos文件（带路径）

edata-BDI-02 report\_home/bin> ./start.sh -t /home/bdi/report\_home/config/report/partner\_acc\_report.xml -s 20131231000000 -u game/hadoop –k game.keytab

1. 执行结束后，可以到HBase中查看结果

get 't\_game\_package\_d','projectid@1001205#relchannel@123.145#gameid@20160216002#dtime@20160414#'

**5. 报表数据查询功能**



代码配置库：<https://szxsvn06-ex:3690/svn/AS_WAD_SVN/trunk/eData/CODE/Baseline/edp>

1. 页面在edp的基础上开发，在\edp\WebContent\eData\game开发相应的页面（入口页面参考：dashboard-project-situation-single页面，这次开发是将相关的页面迁移到game目录下），页面中调用Portal对象提供的方法，从后台查询数据（JSon格式返回），需要引入支持如下方法：

getJson ------------------  查询数据，参数为查询逻辑的Key值（如示例中的overall\_recent\_detail）。

getJsonWithCondition  ------------------  根据指定的条件查询数据，第一个参数为查询逻辑的Key值，第二个参数为查询条件（JSon格式，Key-Value）

getJsonWithArgCurrentUser  ------------------  根据指定的条件查询当前登录用户的信息，登录用户名在SSO中同步到会话中，第一个参数为查询逻辑的Key值，第二个参数为查询条件（JSon格式，Key-Value）。

样例代码如下

Portal.getJsonBySpWithCondition("overall\_recent\_detail","{'DATETIME':'"+datetime+"','DATETYPE':'"+datetype+"'}",

                               function(qResult){

                   if(qResult==null)

                   {

                            return;

                   }

                   var obj = JSON.parse(qResult);

                   var TIME = obj.TIME;

                   var trendTime = new Array();

                   if (TIME instanceof Array){

                            trendTime = TIME;

                   }

                   //一条数据的 后台未以数组返回

                   else{

                            trendTime[0] = TIME;

                   }

                   //具体的赋值语句

                   trendDetailData = initArray();

                   for (var k = 0;k < 9;k++){

                            trendDetailData[k]=new Array();

                            var trendDisplay = overallDetailTrend[k];

                      for(var i = 0;i < trendTime.length;i++)

                            {

                                     labelValue=trendTime[i];

                                     trendDetailData[k][i] = {

                                             value: [obj[trendDisplay][i]],

                                             label: labelValue,

                                             attr1: trendTime[i]};

                            }

                   }

                   //list data

                   listDetailData = new Array();

                   var collen = trendDetailData.length + 1;

                   for (i = 0;i < trendTime.length;i++){

                            listDetailData[i] = new Array();

                            listDetailData[i][0] = trendTime[i];

                            for( j = 1 ;j < collen; j++){

                                     listDetailData[i][j] = obj[overallDetailTrend[j-1]][i];

                            }

                   }

                   var needRow = (Math.ceil(listDetailData.length/12))\*12;

                   if (listDetailData.length > 0)

                   {

                            addTableRows('overall-overview-detail',needRow,listDetailData[0].length);

                   }

                   else

                   {

                            addTableRows('overall-overview-detail',12,10);

                   }

                   insertTableData('overall-overview-detail',listDetailData,displayDetailTrend);

                   addOperation('overall-overview-detail',obj);

                   addPageDiv('pageSplit',needRow,12);

                   listSplit('overall-overview-detail',12);

         }

1. 服务器提供封装好的PortalServlet（本次开发应该不需要扩展），根据web.xml的配置来引用RMI远程DM模块（由edp框架提供）

<servlet>

        <servlet-name>ReportServlet</servlet-name>

        <servlet-class>com.huawei.incloud.edata.servlet.webapp.servlet.ReportServlet</servlet-class>

        <init-param>

            <param-name>fetcherURL</param-name>

            <param-value>//localhost:8099/DataFetcher</param-value>

        </init-param>

        <init-param>

            <param-name>modifierURL</param-name>

            <param-value>//localhost:8099/DataModifier</param-value>

        </init-param>

    </servlet>

1. DM启动 读取预先开发的配置文件（本次开发新增的配置需要存放在dm模块的config目录下），在收到客户端调用请求时，根据查询逻辑的Key值获得对应的处理逻辑，解析相应的参数进行处理，支持如下两种查询逻辑

3.1  procedurceFetcher ------------------- 调用Oracle的存储过程查询数据

一个配置文件中只能有一个dataFetchers和一个dbFetcherGroup，可以有多个procedurceFetcher。

dataSrcName为数据源的名称（在etc/dbconf.xml中定义）。

resultType为返回值的格式，支持如下四种取值

singleMap ---- {k1:v1,k2:v2,……}

arrayMap ---- {a:[],b:[],c:[]}

arrayArray ----  [[],[],[]]

mapArray  ----   [{},{},{}]

name 为 查询逻辑的Key值，全局唯一，如果存在冲突，后加载的将被丢弃。

procedure为存储过程的名称，最后一个参数返回的游标。

parameter为存储过程需要的参数，name为请求中携带的参数名，type为参数类型，支持NUMBER和STRING两种类型。

<dataFetchers>

    <dbFetcherGroup>

        <dataSrcName>edata</dataSrcName>

        <!-- arrayMap   {"DATES":[0,0,0,0],"DOMAIN":["API","Game","Video","M2M"]} -->

        <!-- singleMap  {"DATES":0,"DOMAIN":"API"} -->

                   <!--项目分析-内容概况  近日数据-->

<procedurceFetcher> <!-- 1 -->

            <resultType>singleMap</resultType>

            <name>project\_content\_num</name>

            <procedure>p\_project\_content\_num(?,?,?,?)</procedure>

            <parameter>

                   <name>PROJECTID</name>

                   <type>STRING</type>

            </parameter>

            <parameter>

                   <name>STARTTIME</name>

                   <type>STRING</type>

            </parameter>

            <parameter>

                   <name>ENDTIME</name>

                   <type>STRING</type>

            </parameter>

        </procedurceFetcher>

         </dbFetcherGroup>

</dataFetchers>

3.2  fetcher------------------- 从HBase中查询数据

一个配置文件中只能有一个dataFetchers和一个hBaseFetcherGroup，可以有多个fetcher。

hBaseFetcherGroup.: user 连接HBase的用户名，keyTab 连接HBase的Kerbos文件，默认是在dm\_home目录下。

fetcher ：

resultType 返回类型，同procedurceFetcher，目前支持singleMap和arrayMap

name 为 查询逻辑的Key值，全局唯一，如果存在冲突，后加载的将被丢弃。由于本次开发主要是统计逻辑的变动，页面复用，因此name可以采用基础版本中procedurceFetcher同名的name，原配置文件中相应的配置需要注释掉。

From 为HBase中需要查询的表。

Scope 为 查询的范围，支持=和between判断条件，支持加/减算式。以空格分隔，支持变量，变量名以{}包围，需要在fieldDefinition定义类型，目前只支持日期类型。

keyField为查询的条件字段，name为HBase表中的字段名，paramName为客户传送的参数名，可以包含多个keyField

resultField为返回结果集中需要包含的字段名，多个字段名以逗号分隔，需要在fieldDefinition定义类型。

fieldDefinition中可以包含多个field。

Field ： name 为 HBase中的字段名。必填。

type为类型，支持DATE、int、double。必填

format 为 显示的格式，可以为空。

alias 为返回结果集中的名称，可以为空，缺省情况下使用name

formula 为数据转换的公式，支持?:表达式，支持四则运算，支持函数调用（函数名以$开头，V\_开头的为客户端携带的参数，非V\_开头的为字段名，FETCHERGROUP为内置的对象变量，提供查询HBase的能力）

<dataFetchers>

    <hBaseFetcherGroup user="game/hadoop" keyTab="conf/game.keytab">

                   <!-- 单个内容昨日概况-->

                   <fetcher>

                            <resultType>singleMap</resultType>

                            <name>single\_content\_recent</name>

                            <from>t\_content\_summary\_d</from>

                            <scope expression = "dtime = {ENDTIME} - 1" />

                            <keyField name = "projectid" paramName="PROJECTID"/>

                            <keyField name = "contentid" paramName="CONTENTID"/>

                            <keyField name = "relchannel" paramName="CHANNELID"/>

                            <resultField>

                            downloadcnt,startcnt,newusercnt,activeusercnt,totalusercnt,payamount,downloadusercnt,payusercnt,totalpayamount,perpayamount,paycnt,arpu,arppu,avgusecnt,avrusedur,payrate

                            </resultField>

                            <fieldDefinition>

                                     <field name = "dtime" type = "DATE" format = "yyyyMMdd" alias="TIME"/>

                                     <field name = "downloadcnt" type = "int"/>

                                     <field name = "startcnt" type = "int"/>

                                     <field name = "newusercnt" type = "int"/>

                                     <field name = "activeusercnt" type = "int"/>

                                     <field name = "accusercnt" type = "int" alias="totalusercnt"/>

                                     <field name = "payamount" type = "double" formula="$convertMoney(V\_LANG,FETCHERGROUP,PAYAMOUNT)"/>

                                     <field name = "downloadusercnt" type = "int"/>

                                     <field name = "payusercnt" type = "int"/>

                                     <field name = "accpayamount" type = "double" alias="totalpayamount"

                                                  formula="$convertMoney(V\_LANG,FETCHERGROUP,ACCPAYAMOUNT)"/>

                                     <field name = "paycnt" type = "double"/>

                                     <field name = "avrusedur" type = "double"/>

                                     <field name = "perpayamount" type = "double"

                                                  formula="PAYAMOUNT > 0?(PAYCNT > 0 ? $convertMoney(V\_LANG,FETCHERGROUP,PAYAMOUNT)/PAYCNT: -1):0"/>

                                     <field name = "arpu" type = "double"

                                                  formula="PAYAMOUNT > 0?(ACTIVEUSERCNT > 0 ?$convertMoney(V\_LANG,FETCHERGROUP,PAYAMOUNT)/ACTIVEUSERCNT: -1):0"/>

                                     <field name = "arppu" type = "double"

                                                  formula="PAYAMOUNT > 0?(PAYUSERCNT > 0 ?$convertMoney(V\_LANG,FETCHERGROUP,PAYAMOUNT)/PAYUSERCNT : -1):0"/>

                                     <field name = "avgusecnt" type = "double"

                                                  formula="STARTCNT > 0?(ACTIVEUSERCNT>0?STARTCNT/ACTIVEUSERCNT:-1):0"/>

                                     <field name = "payrate" type = "double"

                                                  formula="PAYUSERCNT > 0?(ACTIVEUSERCNT>0?PAYUSERCNT/ACTIVEUSERCNT:-1):0"/>

                                     <field name = "{ENDTIME}" type = "DATE" format = "yyyy-MM-dd" />

                            </fieldDefinition>

                   </fetcher>

         </hBaseFetcherGroup>

</dataFetchers>

3.3  pageFetcher------------------- 从HBase中分页列表查询数据，继承自fetcher

rowKey : HBase中RowKey的格式定义，用来实现根据主键进行条件过滤，主键的顺序和HBase中RowKey保持一致。主键之前以#分隔，键值之间以@分隔。

orderby为列表显示时的排序字段。

<dataFetchers>

    <hBaseFetcherGroup user="game/hadoop" keyTab="conf/game.keytab">

                   <!-- 单个内容昨日概况-->

                            <resultType>arrayMap</resultType>

                            <name>overall\_recent\_detail</name>

                            <from>t\_content\_summary\_d</from>

                   <rowKey>[projectid@{projectid}#contentid@[0-9a-zA-Z./-]\*#relchannel@[0-9a-zA-Z./-]\*#dtime@{dtime}#version@[0-9a-zA-Z./-]\*#</rowKey](mailto:projectid@%7bprojectid%7d#contentid@[0-9a-zA-Z./-]*#relchannel@[0-9a-zA-Z./-]*#dtime@{dtime}#version@[0-9a-zA-Z./-]*#</rowKey)>

                            <scope expression = "dtime = {DATETIME} - 1" />

                            <orderBy>downloadcnt</orderBy>

                            <filterField name = "projectid" paramName="QUERYPROJECTID" isKey="true"/>

                            <filterField name = "relchannel" paramValue="ALL" isKey="true"/>

                            <filterField name = "version" paramValue="ALL" isKey="true"/>

                            <filterField name = "partnerid" paramName='PARTNERID==""?QUERYPARTNERID:PARTNERID'/>

                            <resultField>

                            time,partnerid,contribution,startcnt,contentid,appkey,projectcnt,downloadcnt,cpname,newusercnt,payamount,activeusercnt,payusercnt,totalusercnt,extcontentid

                            </resultField>

                            <fieldDefinition>

                                     <field name = "dtime" type = "DATE" format = "yyyyMMdd"/>

                                     <field name = "contentname" type = "string"  alias="time" nullable="false"/>

                                     <field name = "partnername" type = "string"  alias="cpname" nullable="false"/>

                                     <field name = "downloadcnt" type = "int"/>

                                     <field name = "projectcnt" type = "int"/>

                                     <field name = "startcnt" type = "int"/>

                                     <field name = "newusercnt" type = "int"/>

                                     <field name = "activeusercnt" type = "int"/>

                                     <field name = "accusercnt" type = "int" alias="totalusercnt"/>

                                     <field name = "payamount" type = "double" formula="$convertMoney(V\_LANG,FETCHERGROUP,PAYAMOUNT)" format="0.00"/>

                                     <field name = "payusercnt" type = "int"/>

                                     <field name = "contribution" type = "double" format="0.00"/>

                                     <field name = "partnerid" type = "string"/>

                                     <field name = "contentid" type = "string"/>

                                     <field name = "appkey" type = "string"/>

                                     <field name = "extcontentid" type = "string"/>

                                     <field name = "{DATETIME}" type = "DATE" format = "yyyy-MM-dd" />

                            </fieldDefinition>

                   </pageFetcher>

         </hBaseFetcherGroup>

</dataFetchers>

**6. 调度流开发**

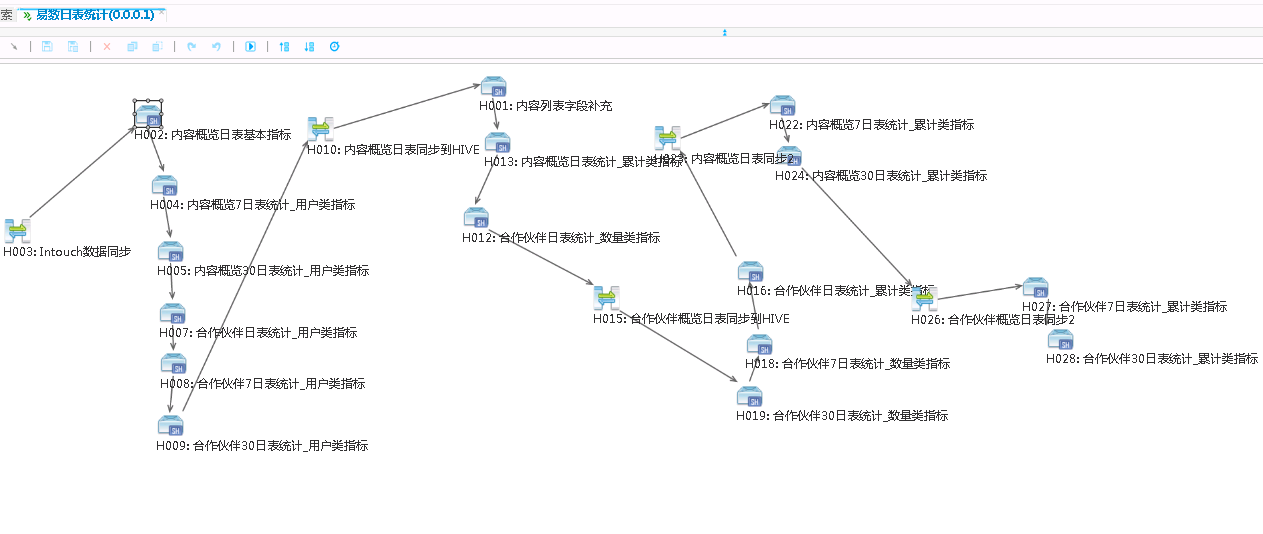
在完成单个报表的开发后，需要通过BDI将整个流程串联和调度。BDI中通过外部程序执行报表的统计。BDI中需要考虑数据的依赖关系，步骤之间串行选择（统计模块中已考虑指标的并行处理）。

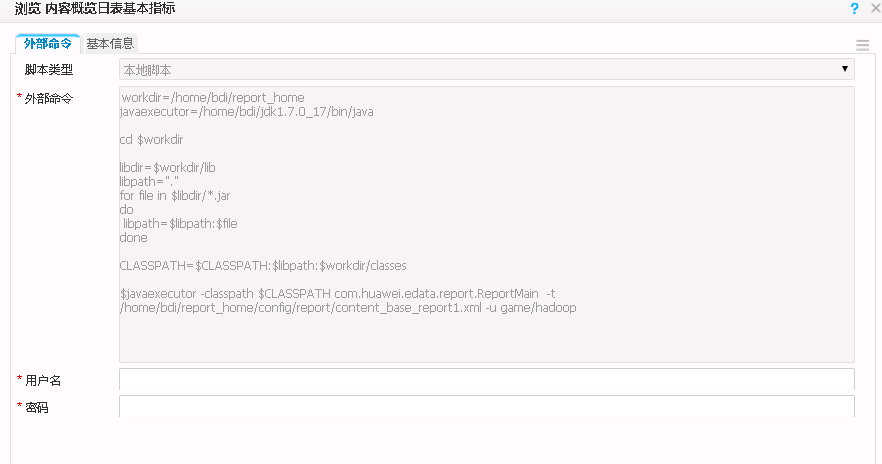
使用BDI的抽取和加载功能实现HBase统计结果向Hive的同步，作为后续数据统计的基础表。

对于部分不需要页面呈现的中间结果表，如果本身比较简单的话（不超过三步），也可以直接使用BDI的计算功能（比如：抽取+转换+加载）进行处理，导入到HIVE库中供数据统计算使用（比如：记录用户行为的表）。

由于HBase本身不支持表间的关联查询，如果在数据统计时将表间关联的字段合并到主表中（比如下面的内容列表字段补充，实现的是将内容名称、合作伙伴名称等字段加入到内容概览表中。

内容统计的调度流如下，项目分析调度流可以参考：





workdir=/home/bdi/report\_home

javaexecutor=/home/bdi/jdk1.7.0\_17/bin/java

cd $workdir

libdir=$workdir/lib

libpath="."

for file in $libdir/\*.jar

do

libpath=$libpath:$file

done

CLASSPATH=$CLASSPATH:$libpath:$workdir/classes

$javaexecutor -classpath $CLASSPATH com.huawei.edata.report.ReportMain  -t /home/bdi/report\_home/config/report/content\_base\_report1.xml -u game/hadoop