|  |
| --- |
|  |

|  |
| --- |
| **调度系统表结构说明文档** |
| 统计平台项目组 |
|  |
|  |
|  |
| **Kendy** |
| **2014-9-18** |
|  |

目录

[修改历史记录 1](#_Toc401665529)

[工作记录 2](#_Toc401665530)

[一、数据库表结构 2](#_Toc401665531)

[1、 hadoop任务信息表 t\_job\_mapreducer 2](#_Toc401665532)

[2、 shell or php任务信息表 t\_job\_shphp(根据文件名后缀区分是shell文件或php文件) 3](#_Toc401665533)

[3、 DB\_UPLOAD任务信息表 t\_job\_db\_upload 3](#_Toc401665534)

[4、 MYSQL\_UPLOAD任务信息表 t\_job\_mysql\_upload 4](#_Toc401665535)

[5、 hadoop任务参数表 t\_job\_parameter 4](#_Toc401665536)

[6、 hadoop任务调度表 t\_job\_process 4](#_Toc401665537)

[7、 hadoop任务分类表 t\_job\_classification 5](#_Toc401665538)

[二、调度处理逻辑 5](#_Toc401665539)

[1、前期准备工作 5](#_Toc401665540)

[2、调度处理逻辑 5](#_Toc401665541)

[3、数据间隔符说明 6](#_Toc401665542)

[三、需要注意的地方 7](#_Toc401665543)

[四、错误码 7](#_Toc401665544)

[五、修数据 8](#_Toc401665545)

修改历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 修改人 | 颜色 | 备注 |
| 2014-09-18 | Kendy | 黑色 | 新建 |
| 2014-09-19 | Kendy | 橙色 | 1.hadoop任务信息表中outKey、outFormat、  inFormat、outValue 四个字段去掉相同的部分  2.hadoop任务调度表中添加自增字段id |
| 2014-09-22 | Kendy | 紫色 | hadoop任务信息表中添加循环字段（cycle） |
| 2014-09-23 | Kendy | 浅蓝 | 1.DB\_UPLOAD任务信息表中将task\_id改为varchar类型，因为有些task\_id以表达式形式表示  2.增加对间隔符的说明 |
| 2014-09-25 | Kendy | 蓝色 | 增加hadoop任务分类表(添加START和END类型) |
| 2014-10-14 | Kendy | 绿色 | 梳理整个逻辑，添加一些说明 |
| 2014-10-17 | Kendy | 水绿 | 生成job产生的错误码整理 |
| 2014-10-20 | Kendy | 深红 | 1. 执行任务失败退出时，会产生错误码（exit num） 2. 增加分游戏策略（MR表中添加gameInfo字段） 3. MR表中根据分游戏对数据形式进行更改 |
| 2014-10-21 | Kendy | 红色 | 1.每种错误码的添加  2.根据传入的gameid生成相应的文件，并对源数据进行覆盖  3.根据传入的jobid，返回jobname |
| 2014-10-21 | Kendy | 红色 | Hadoop任务信息表中添加字段fix\_flag，用于修数据 |
| 2016-03-18 | looper |  | 对线上的t\_job\_process表中的各个process\_id做说明 |

工作记录

|  |  |  |
| --- | --- | --- |
| 日期 | 工作人 | 工作内容 |
| 2014-09-23 | Kendy | 1. 在60上对所有表进行建立。 2. hadoop任务信息表，日报涉及到的job数据填充 3. DB\_UPLOAD任务信息表的所有数据填充。 4. hadoop任务信息表中对上传字段进行更改，以对应DB\_UPLOAD任务信息表。 5. hadoop任务参数表数据的填充（待补充）。 |
| 2014-09-25 | Kendy | 数据库中新建hadoop任务信息表 |
|  |  |  |
|  |  |  |

一、数据库表结构

# hadoop任务信息表 t\_job\_mapreducer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | KEY | NULL | 默认值 | 说明 |
| id | int(11) |  | NO | auto\_increment | 自增 ID |
| job\_id | int(11) | PRI | NO |  | Job ID |
| job\_name | varchar(255) |  | NO | ${date} | Job 名称 |
| gameInfo | varchar(255) |  | NO | ${GAMEINFO} | Gameid信息（分游戏） |
| jar | varchar(255) |  | NO | ${HADOOP\_JAR\_PATH} | Jar 包位置 |
| params | text |  | NO | ‘’ | -D参数 |
| conf | varchar(255) |  | NO | ${HADOOP\_CONF} | conf文件位置 |
| outKey | varchar(255) |  | NO | Text | 输出key类型 |
| outValue | varchar(255) |  | NO | Text | 输出value类型 |
| inFormat | varchar(255) |  | NO | TextInputFormat | 输入格式化类型 |
| outFormat | varchar(255) |  | NO | TextOutputFormat | 输出格式化类型 |
| inputPath | text |  | NO | ‘’ | 输入路径和mapper |
| outputPath | varchar(255) |  | NO | ‘’ | 输出路径 |
| combinerClass | varchar(255) |  | NO | ‘’ | combiner类 |
| reducerClass | varchar(255) |  | NO | ‘’ | reducer类 |
| mos | text |  | NO | ‘’ | 多输出 |
| dbUpload\_id | varchar(128) |  | NO | ‘’ | DB\_UPLOAD 上传ID |
| mysqlUpload\_id | varchar(128) |  | NO | ‘’ | MYSQL\_UPLOAD 上传ID |
| fix\_flag | tinyint(4) |  | NO | 0 | 修数据标识  0：可以分游戏 后缀G\*  1：可以分游戏 后缀/\*/\*\_basic  2：可以分游戏 后缀/\*/\*\_custom  3.：不可以分游戏 |

**例子：**

|  |  |  |  |
| --- | --- | --- | --- |
| job\_id | job\_name | jar | Params |
| 1 | Basic $date | ${HADOOP\_JAR\_PATH} | -D divide=$divide |

|  |  |  |
| --- | --- | --- |
| conf | outKey | outValue |
| ${HADOOP\_CONF} | org.apache.hadoop.io.Text | org.apache.hadoop.io.Text |

|  |  |
| --- | --- |
| inFormat | outFormat |
| org.apache.hadoop.mapred.TextInputFormat | org.apache.hadoop.mapred.TextOutputFormat |

|  |  |
| --- | --- |
| inputPath | outputPath |
| ${RAW\_DIR}/$date/\*/\*\_basic, com.taomee.bigdata.basic.BasicMapper | ${DAY\_DIR}/$date/basic |

|  |  |
| --- | --- |
| combinerClass | reducerClass |
| com.taomee.bigdata.basic.BasicCombiner | com.taomee.bigdata.basic.BasicReducer |

|  |  |  |
| --- | --- | --- |
| mos | dbUpload\_id | mysqlUpload\_id |
| -addMos "UCOUNT,\*\*\* " -addMos "COUNT,\*\*\* " |  |  |

注：若dbUpload\_id或mysqlUpload\_id不为空，表示成如下形式：120,121,122

对应DB\_UPLOAD任务信息表或MYSQL\_UPLOAD任务信息表的upload\_id字段。

# shell or php任务信息表 t\_job\_shphp(根据文件名后缀区分是shell文件或php文件)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | KEY | NULL | 默认值 | 说明 |
| id | int(11) |  | NO | auto\_increment | 自增ID |
| job\_id | int(11) | PRI | NO |  | Job ID |
| file\_name | varchar(128) |  | NO | ‘’ | 文件名 |
| params | text |  | NO | ‘’ | 命令参数 |
| log\_path | varchar(255) |  | NO | ‘’ | log存放目录 |

**例子：**

|  |  |  |  |
| --- | --- | --- | --- |
| shell\_id | file\_name | params | log\_path |
| 1 | \*\*\*.sh | $date true | $WORKDIR/log/$date/\*\*\*.log |
|  | | | |
|  | | | |

# DB\_UPLOAD任务信息表 t\_job\_db\_upload

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | KEY | NULL | 默认值 | 说明 |
| id | int(11) |  | NO | auto\_increment | 自增ID |
| upload\_id | int(11) | PRI | NO |  | 上传任务ID |
| type | tinyint(4) |  | NO | 2 | 上传类型 |
| time | varchar(128) |  | NO | $date | 上传日期 |
| task\_id | varchar(128) |  | NO | -1 | 加工ID |
| path | varchar(255) |  | NO | ‘’ | 上传路径 |

**例子：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| upload\_id | type | time | task\_id | path |
| 1 | 2 | $date | 67 | ${SUM\_DIR}/$date/pay-all-ucount/part\* |
|  | | | | |
|  | | | | |

注：若task\_id为-1，则输出不会带“-task”参数，否则带“-task”参数。

# MYSQL\_UPLOAD任务信息表 t\_job\_mysql\_upload

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | KEY | NULL | 默认值 | 说明 |
| id | int(11) |  | NO | auto\_increment | 自增ID |
| upload\_id | int(11) | PRI | NO |  | 上传ID |
| table\_name | varchar(128) |  | NO | ‘’ | 需要操作的表名 |
| table\_field | text |  | NO | ‘’ | 需要操作的表字段 |
| path | varchar(255) |  | NO | ‘’ | 上传路径 |
| time | varchar(128) |  | NO | $time | 上传日期 |

**例子：**

|  |  |  |
| --- | --- | --- |
| upload\_id | table\_name | table\_field |
| 1 | t\_item\_sale\_data | "sstid,game\_id,platform\_id,zone\_id,server\_id,item\_id,vip,salenum,salemoney,buycount,buyucount" |

|  |  |
| --- | --- |
| path | time |
| ${SUM\_DIR}/$date/buyitem/part\* | $time |
|  | |

# hadoop任务参数表 t\_job\_parameter

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | KEY | NULL | 默认值 | 说明 |
| param\_id | int(11) | PRI | NO | auto\_increment | 参数ID |
| param\_name | varchar(255) |  | NO | ‘’ | 参数名称 |
| param\_content | varchar(255) |  | NO | ‘’ | 参数内容 |

**例子：**

|  |  |  |
| --- | --- | --- |
| param\_id | param\_name | param\_content |
| 1 | $date | $1 |
| 2 | $yesterday | `data -d "$date -1 day" +%Y%m%d` |
| 3 | $nday | `date -d "$date -${n} day" +%Y%m%d` |
|  | | |

# hadoop任务调度表 t\_job\_process

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | KEY | NULL | 默认值 | 说明 |
| id | int(11) | PRI | NO | auto\_increment | 唯一标识id |
| process\_id | int(11) |  | NO | -1 | 工作流id |
| job\_id | int(11) |  | NO | -1 | Job ID |
| after\_job | varchar(128) |  | NO | ‘’ | 后续需要处理的job |

**例子：**

|  |  |  |  |
| --- | --- | --- | --- |
| id | process\_id | job\_id | after\_job |
| 1 | 1 | 0 | 1 |
| 2 | 1 | 1 | 2,3 |
| 3 | 1 | 2 | 4 |
| 4 | 1 | 3 | 4 |
| 5 | 1 | 4 | 5,6,7 |
| 6 | 1 | 5 | end |
| 7 | 1 | 6 | 8 |
| 8 | 1 | 7 | 8 |

# hadoop任务分类表 t\_job\_classification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | KEY | NULL | 默认值 | 说明 |
| job\_id | int(11) | PRI | NO | auto\_increment | 唯一标识job id |
| type | tinyint(4) |  | NO | 0 | job类型标志  0:START 9:END  1:MR  2:shell  3:php  4:DB\_UPLOAD  5:MYSQL\_UPLOAD |

二、调度处理逻辑

# 1、前期准备工作

（1）创建所有的三种表：任务表（MR、shell、php、upload），参数表，调度表。

（2）填充hadoop任务信息表的内容。根据当前加工项脚本，对每一个MapReducer任务的信息进行填充入库，除去一些自带默认值的字段，主要填充以下字段：job\_name、params、inputPath、outputPath、combinerClass、reducerClass、mos，其中inputPath的填充形式为：输入路径+Mapper。最后，一些MR任务结束后有入库操作，此时需要对dbUpload\_id或者mysqlUpload\_id字段进行填充。

（3）填充shell or php任务信息表的内容。其中shell任务需要在第二步MR任务文件生成的前提下，进行填充。

（4）填充DB\_UPLOAD任务信息表。分两种形式：带task\_id与不带task\_id。当入库语句不带task\_id参数时，不需要填task\_id字段，默认值-1即代表不带task\_id。

（5）填充MYSQL\_UPLOAD任务信息表。

（6）填充hadoop任务参数表。

（7）填充hadoop任务调度表。其中由process\_id字段区分工作流，同一个工作流的process\_id字段相等。需要根据输入输出，对所有任务的衔接关系进行归类，以此填充after\_job字段。同时该表需要设置开始节点0和结束节点end。

# 2、调度处理逻辑

（1）工作流中的任务主要有四种处理方式：

a．顺序执行：

A=>

B

C

D

A=>B=>C

b．一个任务执行完后多个任务执行：

=>E

B

C

D

c．多个任务全部执行完后，执行下一个任务:

=>E

B

C

D

=>F

d．多个任务部分执行完后，执行下一个任务：

（2）任务调度表的工作方式，以下面数据为例。

|  |  |  |
| --- | --- | --- |
| process\_id | job\_id | after\_job |
| 1 | 0 | 1 |
| 1 | 1 | 2,3 |
| 1 | 2 | 4 |
| 1 | 3 | 4 |
| 1 | 4 | 5,6,7 |
| 1 | 5 | end |
| 1 | 6 | 8 |
| 1 | 7 | 8 |

1. 这是一条工作流，工作流id为1
2. 起始节点为job0，然后执行第一个任务job1
3. job1执行完后，并行执行job2、job3
4. 当job2、job3全部执行完后，执行job4
5. job4执行完后，并行执行job5、job6、job7
6. job5执行完后，后续没有衔接其他任务，走到end节点
7. 当job6、job7全部执行完后，执行job8
8. 后续依次类推，直到全部走到end节点。

# 3、数据间隔符说明

（1）mos

每个mos中的四个字段以“:”间隔：mos名字：输出格式化类型：输入类型：输出类型。多个mos之间以“|”间隔。例如：



（2）input

多个input之间以“|”间隔。例如



注：有的input最后一个字段为“0”，代表该输入可能没有，需要判断。

（3）上传字段均以“|”间隔，例如



（4）循环字段均以“|”间隔，例如



（5）参数均以“|”间隔，例如



4. 调度处理流程

（1） hadoop任务调度表t\_job\_process中获取一个job\_id

（2） 根据job\_id去hadoop任务分类表中获取该job的类型

（3） 根据job\_id和job类型去查询对应的表，生成相应的文件，例如：XXX.sh XXX.php

（4） 执行生成的文件

（5） hadoop任务调度表t\_job\_process中获取下一个job\_id …依次类推

三、需要注意的地方

1．任务数据进行插表的时候先对任务分类表进行填充，然后根据job\_id 去填充相应的其他表。

2. 任务表中除了有自增字段还有job\_id字段，该job\_id字段需要根据任务分类表进行设定。

3. Mapreduce任务不存在循环任务，若有循环，需要分开为单个任务

4. 所有参数统一格式，一律表示成${XXX}的个格式

5. 时间参数统一格式：

本天 ${date} 昨天之前 ${last\_1\_day} ${last\_2\_day} ${last\_3\_day}

本周 ${this\_week} 上周之前 ${last\_1\_week} ${last\_2\_week} ${last\_3\_week}

本月 ${this\_month} 上月之前 ${last\_1\_month} ${last\_2\_month} ${last\_3\_month}

1. 参数表中的数据更改了形式

原来 参数名：date 参数内容：$1

现在 参数名：date 参数内容：date=$1

更改原因：防止有的变量是由多个表达式计算出来，例如this\_Monday

1. 对于像$XXX=$1 $YYY=$2 这种变量是需要判断的，看有没有传入参数

四、错误码

|  |  |
| --- | --- |
| **任务文件生成过程中产生的错误码** | **错误信息** |
| 0 | 正常 |
| 1 | 数据库中获取job信息失败 |
| 2 | 查询job信息为空 |
| 3 | 数据库中获取参数信息失败 |
| 4 | 查询参数信息为空 |
| 5 | 数据库中获取DB入库信息失败 |
| 6 | 查询DB入库信息为空 |
| 7 | 数据库中获取MYSQL入库信息失败 |
| 8 | 查询MYSQL入库信息为空 |
| 9 | 文件名后缀不匹配 |
| 10 | 打开文件失败 |
| 11 | 写文件失败 |
| 12 | input格式有误，路径最后两个字符不是G\*（用于修数据） |
| 13 | 数据库中获取job名字失败 |
| 14 | 查询job名字为空 |

|  |  |
| --- | --- |
| **任务执行过程中产生的错误码** | **错误信息** |
| 0 | 正常 |
| 1 | 任务执行失败（MR SH PHP） |
| 2 | 日期参数为空 |
| 3 | 输入路径inputs为空 |

五、修数据

1.根据传入的gameid列表更改输入路径

2.更改输出路劲，将输入路径定为临时存储数据的路径

例：原输出路径为：**${ALL\_DIR}/$date/pay-all**

则修数据时的临时存储路径为：**${ALL\_DIR}/$date/pay-all-game**

1. 将原来的输出路径中的涉及到的游戏数据清除，并将临时存储路径中的数据拷贝到原来的输出路径。（这一步在入库之前）
2. 分游戏修数据时，最后的拷贝逻辑有错，已做修改（20141218）

六、线上process\_id说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Process\_id | 功能 | Job\_id个数 | 备注 | 对应脚本 |
| 1 | 新统计日报老调度process | 94 | 之前日报和非日报数据混在一起调度（目前已经不在使用） | 129机器上的  /opt/taomee/hadoop/  bigdata-code/stat-calc/  start\_daily.sh |
| 2 |  | 54 |  |  |
| 3 |  | 12 |  |  |
| 4 | 月报数据的调度计算 | 12 | 使用中，每月1号18点计算上月月报数据 | 129机器上的  /opt/taomee/hadoop/  bigdata-code/stat-calc/  start\_month\_report.sh |
| 5 | 线上小时数据的调度计算 | 45 | 使用中，工作日每天的11~18点没两个小时执行一次 | 129机器上的  /opt/taomee/hadoop/  bigdata-code/stat-calc/  start\_hours.sh |
| 6 |  | 57 |  |  |
| 7 |  | 20 |  |  |
| 8 |  | 10 | 调度里面都是广告的job\_id |  |
| 9 |  | 17 |  |  |
| 10 |  | 8 |  |  |
| 11 | 当前线上日报数据的计算 | 43 | 使用中，每天凌晨2点钟开始执行 | 129机器上的  /opt/taomee/hadoop/  bigdata-code/stat-calc/  start\_daily\_all.sh |
| 12 | 当前线上非日报的任务的调度 | 87 | 使用中，在日报任务都计算完成后计算 | 129机器上的  /opt/taomee/hadoop/  bigdata-code/stat-calc/  start\_daily\_all.sh |
| 13 |  | 51 |  |  |
| 14 | 对新加入的任务生成测试的process | 41 |  |  |
| 16 | 创想兵团4399服付费数据的修复 | 16 |  | 129机器上的  /opt/taomee/hadoop/  bigdata-code/stat-calc/start\_rep.sh |
| 21 | 修复所有付费数据 | 34 | 当修复的数据不需要job\_2，process\_id=21 | 129机器上的  /opt/taomee/hadoop/  bigdata-code/stat-calc  /start\_daily\_pay\_repair.sh |
| 22 | 修复所有付费数据 | 44 | 当修复的数据需要job\_2，process\_id=22 | 129机器上的  /opt/taomee/hadoop/  bigdata-code/stat-calc  /start\_daily\_pay\_repair.sh |
| 603 |  | 1 |  |  |