浙江萧山云平台转码及打包接入文档

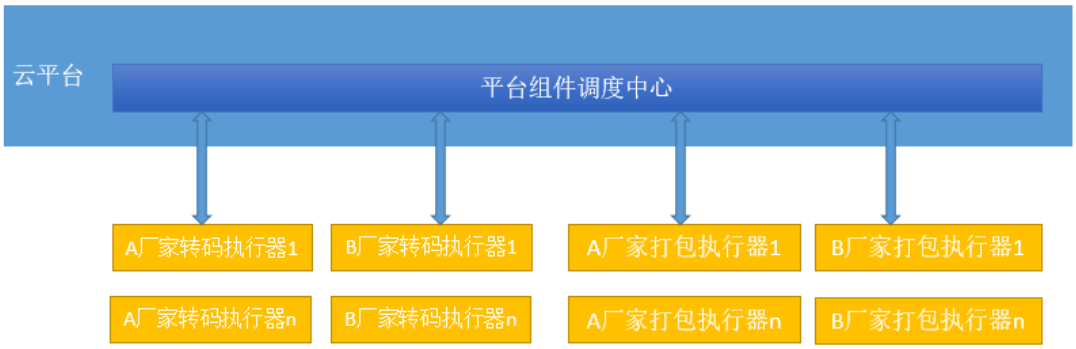
2016-8-15

文档修改记录

编号	版本号	日期	说明	修改者	审核者
1	V1.0	2016-06-28	文档创建	孙永宝、孙翔	钱永江
2	V1.10	2016-07-22	增加不同厂家的 jobType 描述	孙永宝、肖雄	钱永江
3	V1.11	2016-08-15	增加注意事项	孙永宝	钱永江

目录

第一章 系统结构



平台负责调度，每个厂家的转码和打包执行器需要注册到平台调度中心，接收平台调度中心下发的转码或打包任务，并把任务状态反馈给调度中心。接下来从执行器注册及协议说明两个方面来描述执行器接入云平台的过程。

第二章 执行器注册

第1节 Executor 接入

2.1.1 Executor 注册

当有了各种的 Actor 执行器后,就需要在系统中注册执行器的实例了,也就是 ActorInstance。目前,调度平台暂时支持手动的方式注册执行器的实例。同样,我们提供了 RestAPI 以及管理界面的方式添加执行器的实例：

/v1/cdplb/mpc/instance

作用	增加 ActorInstance 实例
请求类型	POST
返回值	code 200 为成功,否则为失败

请求参数(Body)：

参数	类型	是否必填	说明
actorGUID	String	是	执行器实例的唯一标识
actorTypes	String[]	是	执行器实例可以处理的执行器类型
maxCount	Int	是	最大同时执行数量
timeOut	Int	是	任务超时时间,单位分钟
params	Map<String,String>	否	用于传入执行器的特有的参数,key 和 value 均为字符串
failureTransfer	Int	否	执行器失效后,是否转移正在执行的任务,0 为不转移,1 为转移
heartbeatType	String	否	心跳检测类型,不填表示不进行心跳。passive 被动心跳(推荐),active 主动心跳,ZK zookeeper 心跳
heartbeatIdentity	String	否	被动心跳和 ZK 心跳表示这个执行器的唯一标识,默认等于 actorGUID;主动心跳表示心跳的 URL 地址

应用举例：

POST http：//17.16.131.36:8060/sobeyhive-fp/cdplb/registry

```
{
  "actorGUID"： "XXXXXXXXXXXXXX",
  "actorTypes"： [
    " materialclip"
  ],
  "maxCount"： 1,
  "params"： {"key1"："value1","key2"："value2"},
  "timeOut"： 30000
}
```

2.1.2 执行器任务类型

actorTypes 表示执行器可执行的任务类型，枚举值如下：

materialclip 表示通用转码

etclipadditivedata 表示索贝打包合成

edvclipadditivedata 表示新奥特打包合成

dyclipadditivedata 表示大洋打包合成

如果同一个执行器，如果可执行多种任务，就填写多个值。

2.1.3 executor 反注册

除了注册 executor 外,还允许进行反注册。

/v1/cdplb/mpc/instance/{instanceName}

作用	删除 ActorInstance 实例
请求类型	DELETE
返回值	code 200 为成功,否则为失败

请求参数(PATH)： instanceName 表示要删除的实例名称

2.1.4 executor 交互

调度平台与 executor 的交互依赖了一些规则,需要双方在开发阶段就要遵守。否则调度平台很可能无法与 executor 执行器进行数据的交互(一些特有的我们 Hive 内置了 Adpater 的 executor 不需要)。

2.1.5 onRequest

1. 调度平台与 executor 通过标准的 Restful 进行通信；
2. 双方需要识别对方的 HTTP 头信息；

3. 在调度平台请求 executor 发起任务时,HTTP 头上会带有以下内容：

Content-Type	text/plain;charset=utf-8
MessageID	任务的唯一标识
Priority	优先级　0-999　值越大,优先级越高
ReplyTo	executor 回调调度平台地址
其他	流程上或发起调度任务时,三方请求中的头信息

4. executor 接收到任务后,同步反馈一个 code 为 200 的响应,表示接收到任务；
5. 调度平台会以 POST 的方式请求 executor 的 Rest 接口；
6. 调度平台会把协议以 String 的方式,放入 POST 消息体中。

2.1.6 onFinish

1. 调度平台与 executor 通过标准的 Restful 进行通信；
2. 双方需要识别对方的 HTTP 头信息；
3. 在 executor 回调调度平台时,HTTP 头上需要带有以下内容：

Content-Type	text/plain;charset=utf-8
MessageID	任务的唯一标识
IsFault	任务是否失败
eventType	本次回调的类型, 0　Job 开始通知 2　接收进度更新通知 4　Job 暂停通知 8　Job 暂停恢复通知

	16 Job 删除通知 32 Job 完成通知
errorDescription	如果错误,错误描述
其他	流程上或发起调度任务时,三方请求中的头信息

4. 调度平台接收到消息后,同步反馈一个 code 为 200 的响应;
5. executor 需要以 POST 的方式请求调度平台的 Rest 接口,这个接口的地址在 onstart 时记录在头信息的 ReplyTo 中;
6. executor 需要把反馈协议以 String 的方式,放入 POST 消息体中。

通过上面 onStart 与 onFinish 的交互规范,最少就能保证调度平台与 executor 执行器间能够正确的添加执行任务以及反馈。

第2节 心跳检测

除了增加任务,执行任务,反馈任务外。调度平台还有一个重要的功能就是保证执行器的高可用,在有多台执行器都能做相同的任务的情况下,任意一个 executor 死掉不能影响任务的执行。

这个问题分为了两个方面来看,一个是当任务还没有分发到 executor 前,executor 就挂了。这个处理比较简单,调度平台不会把任务分发给这个已经死掉的 executor 执行。 还有一种情况就是任务已经在 executor 上执行了一半了,然后这个时候 executor 挂掉了。那么,这个时候,就需要配置的策略来告诉调度平台是否把这个任务转发给其他正常的执行器实例从头开始执行,或者继续等待这个死掉的 executor 重新上线并继续执行任务直到任务超时。

但无论是任务分发前挂掉还是分发后挂掉,无论是任务转发还是继续等待。

都需要调度平台即时的感知到 executor 执行器的当前状态。 这就需要心跳检测。

2.2.1 被动检测

所谓被动检测,是站在调度平台的角度来说的。由调度平台提供一个 Restful 的 ping 服务。executor 执行器需要在某个时限周期内调用一次这个 Ping 服务,按照规则是 5S 调用一次。那么当调度平台的 ping 服务接收到这个请求后,根据心跳检测的唯一标识,就知道在当前的时刻这个 executor 执行器是在线的,如果超过两个检测周期还没有接收到 executor 的消息,那么就认为这个 executor 挂掉了,那么就开始执行 ActorInstance 下线的逻辑。

如果要使用这种模式的心跳检测,需要 executor 满足以下接入条件:

1. 调度平台会发布一个 restful 的 ping 服务;
2. executor 需要定时的以 GET 的方式调用这个 ping 服务;
3. ping 服务不接受任何的入参,但是在地址上要求带上心跳唯一标识:
   /v1/cdplb/heartbeat/{instanceIdentity}
4. 当调度平台接收到 executor 发起的 GET 请求后,会同步反馈一个 code200 的状态。

第3节 优先级调整

在系统的运行过程中,有些时候可能会遇到需要调整任务优先级的情况。并且调整优先级又分为两种情况:一种是 executor 支持调整优先级的,那么在调度平台上调整任务优先级,就需要把优先级调整通知到 executor 去。还有一种是 executor 不支持优先级的调整,那么这个使用调整任务的优先级只能是调整任务在调度平台中进行调度的先后顺序,调度平台尽量保证尽快的把任务发送给空闲的 executor 去执行。

如果 executor 支持优先级的调整的话(通常情况下是 executor 能同时执行多个任务,并且能控制 CPU 计算资源的分配的情况下)。那么调度平台需要与 executor 进行交互,这需要满足以下的接入条件：

1. 优先级调整事件以 restful 的方式进行交互；
2. 双方需要识别 HTTP 头信息；
3. 在调度平台请求 executor 发起优先级调整的时候,HTTP 头上会带有以下内容：

Content-Type	text/plain;charset=utf-8
MessageID	任务的唯一标识
Priority	优先级　0-999　值越大,优先级越高
其他	流程上或发起调度任务时,三方请求中的头信息

4. restful 服务不接受任何入参；
5. executor 接收到任务后,同步反馈一个 code 为 200 的响应,表示优先级调整成功；
6. 调度平台会以 POST 的方式调用这个 restful 服务。

第4节 取消任务

除了调整优先级外,有些时候还可能遇到需要取消一个任务的情况。需要注意的是,在目前调度平台的设计中,是没有暂停任务,恢复任务,重新开始任务的说法的。每一个任务都是一个原子态的,只会有一次生命周期。取消掉的任务是不能重做的,如果想要重做,请重新发起任务。

同样,如果 executor 支持取消任务的话。那么调度平台需要与 executor 进行交互,这需要满足以下的接入条件：

1. 任务取消事件以 restful 的方式进行交互；
2. 双方需要识别 HTTP 头信息；
3. 在调度平台请求 executor 发起任务取消的时候,HTTP 头上会带有以下内容：

Content-Type	text/plain;charset=utf-8
MessageID	任务的唯一标识
其他	流程上或发起调度任务时,三方请求中的头信息

4. restful 服务不接受任何入参；
5. executor 接收到任务后,同步反馈一个 code 为 200 的响应,表示任务取消成功；
6. 调度平台会以 POST 的方式调用这个 restful 服务。

第5节 Restful Service API

除了上面所描述到的 RestAPI 外,调度平台还提供了一些 RestAPI,以便对任务进行管理。

2.5.1 /v1/cdplb/start/{taskType}

作用	发起任务
请求类型	POST
返回值	code 200 为成功,否则为失败

请求参数(Header)：

参数	类型	说明
sobeyhive-http-system	String	所属系统
sobeyhive-http-site	String	所属站点
messageid	String	任务的唯一标识
synReturn	boolean	是否同步返回(不调用给 executor 直接返回)
processData	JSON	流程参数
其他	String	其他可能有用的辅助参数

协议以 String 方式放入 POST Body 中。

2.5.2 /v1/cdplb/finish/${messageid}

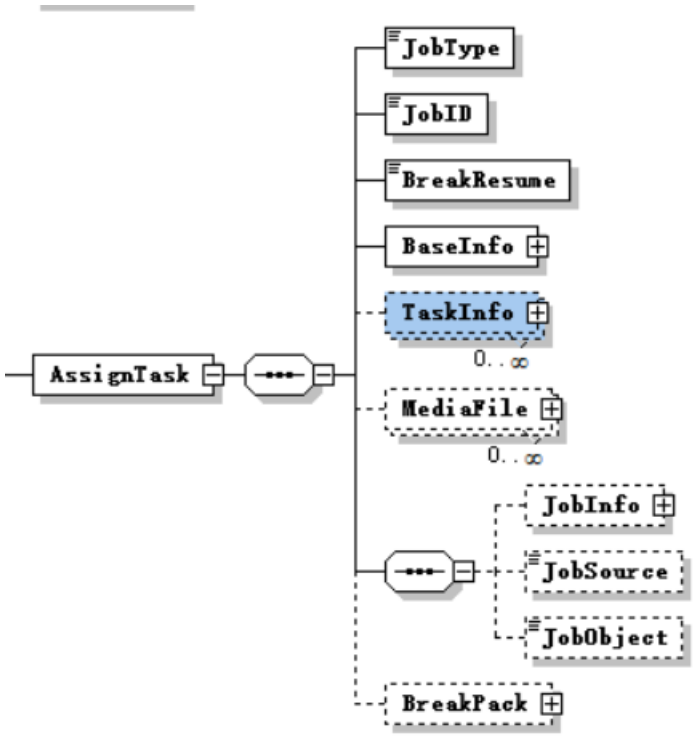
作用	接收任务反馈
请求类型	POST
返回值	code 200 为成功,否则为失败

请求参数(Header)：

参数	类型	说明
messageid	String	任务的唯一标识
IsFault	String	任务是否失败
eventType	String	本次回调的类型，默认为空
errorDescription	String	如果错误,错误描述

第三章 协议说明

第1节 下发到 actor 任务使用协议—AssignTask 协议



3.1.1 主要字段说明

JobType： 任务类型，值如下

materialclip 转码

etclipadditivedata 索贝打包合成

edvclipadditivedata 新奥特打包合成

dyclipadditivedata 大洋打包合成

JobID： 任务 ID 值，任务依据

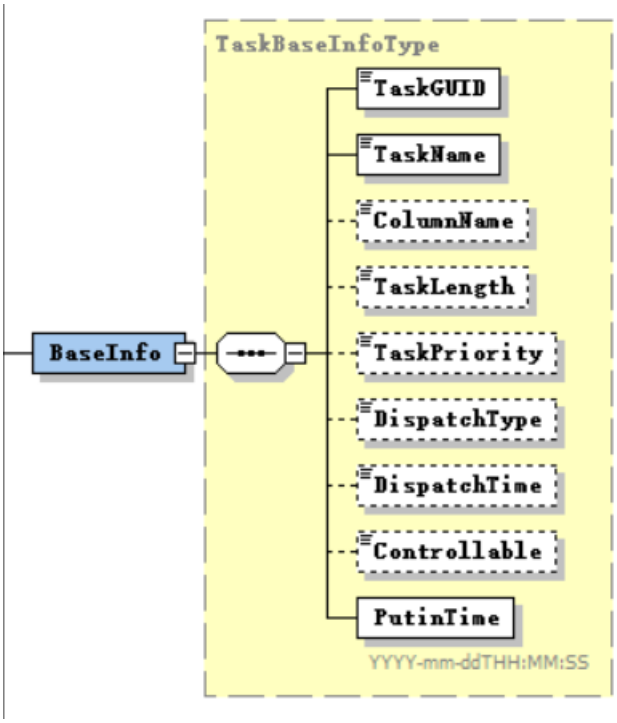
BaseInfo： 任务的一些基础信息

TaskInfo： 附件任务信息

MediaFile： 文件描述信息

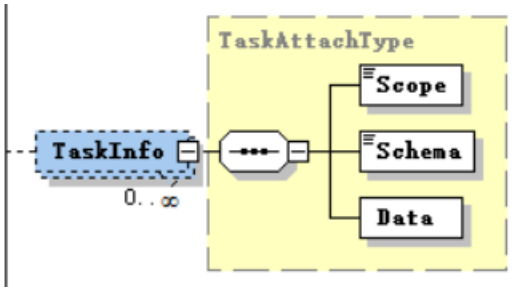
JobInfo： 此次任务信息

具体参考： \转码打包接入协议\转码模板配置参数\baseV2.0.xsd



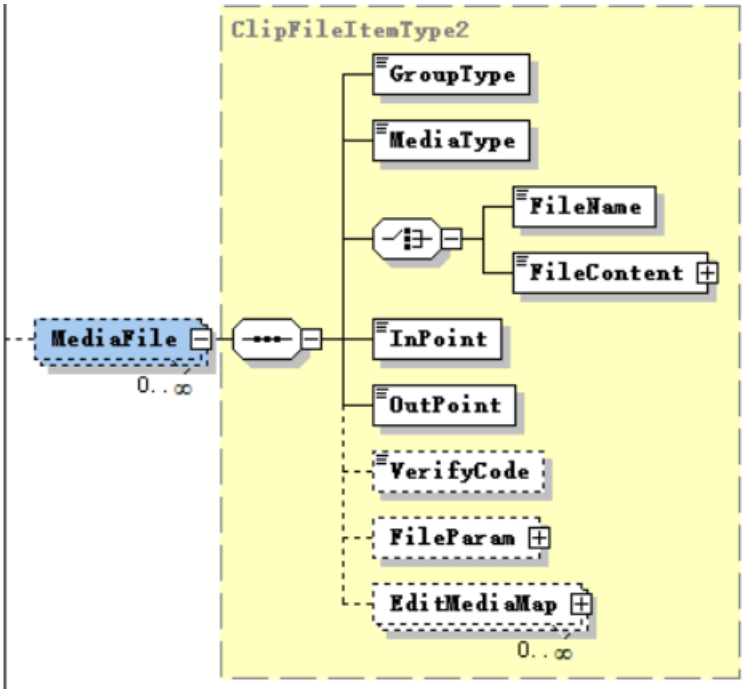
TaskGUID： 任务发送时生成的 guid

TaskName： 任务名



Scope： 附加任务标示

Data： 附加任务具体信息，此处可以有各自厂商自定义，可以作为附加参数传递,比如合成完后的成品素材名称、素材创建者等等信息。



GroupType： 文件组类型

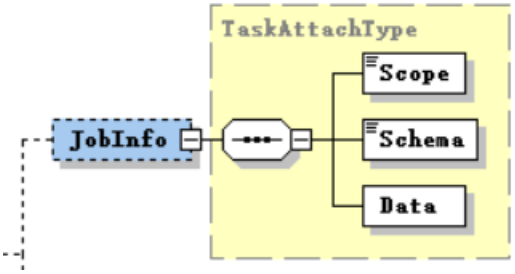
MediaType：¨文件媒体类型

FileName&FileContent： 文件路径描述

InPoint： 文件入点信息，单位百纳秒

OutPoint： 文件出点信息描述，单位百纳秒

FileParam： 文件信息



Scope：

tv_clipmakerjobparam2 转码

etclipadditivedata 索贝打包合成

edvclipadditivedata 新奥特打包合成

dyclipadditivedata 大洋打包合成

Data：　当前任务具体信息

如果是转码，必须为“ClipMakerJobParam2”，打包则是厂商自己定义

3.1.2 样例

```
<MPC>
    <Header>
        <Version>1.1</Version>
        <RequestID>2a4e9fd5a6554b3eb4b627879c90349b</RequestID>
        <RequestTime>20160811 15:59:25</RequestTime>
        <RequestMQ/>
    </Header>
    <Content>
        <MPCType>AssignTask</MPCType>
        <AssignTask>
            <BaseInfo>
                <TaskGUID>a755375c16bb41c7bd763dfd5cd470e3</TaskGUID>
                <TaskName>非编工具</TaskName>
            </BaseInfo>
            <TaskInfo>
                <Scope>DocumentInfo</Scope>
                <Schema/>
                <Data>
                    <DocumentInfo>

    <PGMID>2c91808d567868970156788a7a950001</PGMID>
                        <PGMNAME>非编工具</PGMNAME>
                        <PGMFILE>z:\SobeyInfo\cd02\TLD\非编工具0\非编工具0_0.ntld</PGMFILE>
```

```
                    <PGMLENGTH>28</PGMLENGTH>
                    <CLIPIN>0</CLIPIN>
                    <USERID>352342037</USERID>
                    <ORDERLIST/>
                    <GUIDELENGTH>0</GUIDELENGTH>
                    <GUIDETAIL/>
                    <DOCTAIL/>
                    <DOCID>296</DOCID>
                    <DOCLENGTH>0</DOCLENGTH>
                    <IPADDR>10.20.14.157</IPADDR>
                    <STUDIOSTATUS>PgmPass</STUDIOSTATUS>
                    <COLUMNID>0</COLUMNID>
                    <COLUMNNAME/>
                </DocumentInfo>
            </Data>
        </TaskInfo>
        <MediaFile>
            <GroupType>video</GroupType>
            <MediaType>videogroup</MediaType>
            <FileName>\\\\hive.sobey.com\\hivefiles\\sobeyhive\\bucket-
z\\High-Clip\\0718-mxf测试.mxf</FileName>
            <InPoint>0</InPoint>
            <OutPoint>131600000</OutPoint>
        </MediaFile>
        <JobType> materialclip </JobType>
        <JobID>1</JobID>
        <BreakResume>0</BreakResume>
        <JobInfo>
            <Scope> tv_clipmakerjobparam2 </Scope>
            <Schema>sbclipmakerjobparam2.xsd</Schema>
```

<Data>
< ClipMakerJobParam2>
<Src>
<GroupType>video</GroupType>
<MediaType>videogroup</MediaType>
</Src>
<Option>
<CreateOtcFile>0</CreateOtcFile>
<AnalyzeTarget>1</AnalyzeTarget>
</Option>
<Obj>
<GroupType>dest</GroupType>
<MediaType>FILETYPE_PROXYMP4</MediaType>
<UseTracks>1,2</UseTracks>
<UseAudioMix>0</UseAudioMix>
<PathFormat>//hive.sobey.com/hivefiles/sobeyhive/bucket-p/streamings/2016/07/18/videogroup_0.?ext</PathFormat>
<CodecParam>
<FileFormat>2103</FileFormat>
<VideoFormat>37</VideoFormat>
<BitRate>5120000</BitRate>
<FrameRate>25.0</FrameRate>
<ImageWidth>1280</ImageWidth>
<ImageHeight>720</ImageHeight>
<KeyFrameRate>12</KeyFrameRate>
<Transform>
<Mode>Stretch</Mode>
</Transform>
<AudioFormat>1047</AudioFormat>
<SamplesPerSec>48000</SamplesPerSec>

<BitsPerSample>16</BitsPerSample>

<ReplaceByMainFormat>0</ReplaceByMainFormat>

<AssistFormat>0</AssistFormat>

<SpecialParam><![CDATA[<SpecialParam><VideoParam><sar_width>1</sar_width><sar_height>1</sar_height><speed>6</speed><delay_frames>60</delay_frames><profile>4</profile><level>30</level><definterlace>1</definterlace><interlace>0</interlace><ref_frames>0</ref_frames><bframes>-1</bframes><vbv_buffer>40</vbv_buffer><b_aud>0</b_aud></VideoParam><AudioParam><Bitrate>128000</Bitrate></AudioParam><FileParam><HintTrackValue>0</HintTrackValue><CreateStreamIndex>1</CreateStreamIndex></FileParam></SpecialParam>]]></SpecialParam>

</CodecParam>

</Obj>

</ ClipMakerJobParam2>

</Data>

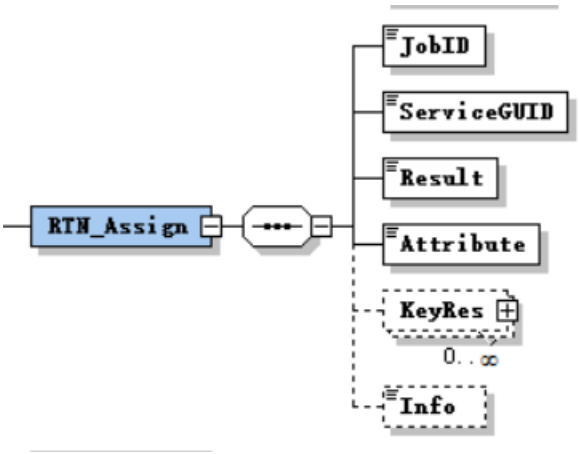
</JobInfo>

</AssignTask>

</Content>

</MPC>

第2节 任务同步反馈协议-- RTN_Assign 协议



3.2.1 主要字段说明

JobID： 任务 ID 值，HIVE 任务依据

ServiceGUID： 任务的 GUID

Result：任务执行结果，accepted 标识任务被成功接受，notsupport 表示任务不被支持，其他的认为任务没有被接受

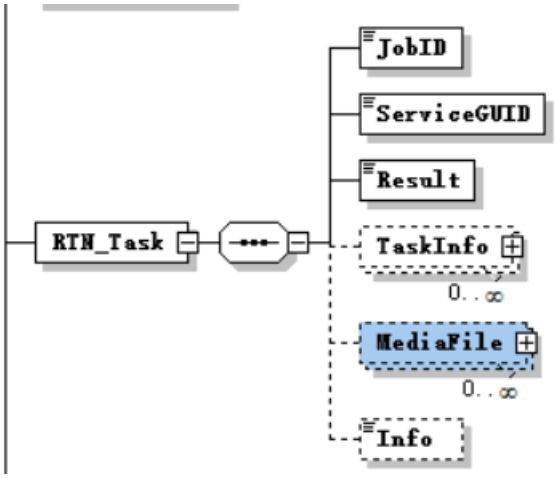
Info： 任务描述信息

3.2.2 样例

```
<MPC>
    <Content>
        <MPCType>RTN_Assign</MPCType>
        <RTN_Assign>
            <JobID>1</JobID>
            <ServiceGUID>eb566725679c48b18daa5a6e0d1a2ba1</ServiceGUID>
            <Result>accepted</Result>
            <Attribute>14</Attribute>
```

```
            <Info>successed</Info>
        </RTN_Assign>
    </Content>
</MPC>
```

第3节 任务完成异步反馈协议--RTN_Task 协议



3.3.1 主要字段说明

JobID： 任务 ID 值，HIVE 任务依据

ServiceGUID： 任务 GUID

Result： 任务执行结果。 succeed 表示任务成功，其他认为失败

TaskInfo： 附件任务信息

MediaFile： 文件描述信息

Info： 任务描述信息

3.3.2 样例

```
<?xml version="1.0" encoding="UTF-8"?>
<MPC>
    <Content>
        <MPCType>RTN_Task</MPCType>
        <RTN_Task>
            <JobID>1</JobID>
            <ServiceGUID>a755375c16bb41c7bd763dfd5cd470e3</ServiceGUID>
            <Result>succeed</Result>
            <TaskInfo>
                <Scope>DocumentInfo</Scope>
                <Schema/>
                <Data>
                    <DocumentInfo>

    <PGMID>2c91808d567868970156788a7a950001</PGMID>
                        <PGMNAME>非编工具</PGMNAME>
                        <PGMFILE>z:\SobeyInfo\cd02\TLD\非编工具0\非编工具
0_0.ntld</PGMFILE>
                        <PGMLENGTH>28</PGMLENGTH>
                        <CLIPIN>0</CLIPIN>
                        <USERID>352342037</USERID>
                        <ORDERLIST/>
                        <GUIDELENGTH>0</GUIDELENGTH>
                        <GUIDETAIL/>
                        <DOCTAIL/>
                        <DOCID>296</DOCID>
                        <DOCLENGTH>0</DOCLENGTH>
                        <IPADDR>10.20.14.157</IPADDR>
                        <STUDIOSTATUS>PgmPass</STUDIOSTATUS>
                        <COLUMNID>0</COLUMNID>
```

```
                        <COLUMNNAME/>
                    </DocumentInfo>
                </Data>
            </TaskInfo>
            <MediaFile>
                <GroupType>video</GroupType>
                <MediaType>videogroup</MediaType>
                <FileName>\\\\hive.sobey.com\\hivefiles\\sobeyhive\\bucket-z\\High-Clip\\0718-mxf测试.mxf</FileName>
                <InPoint>0</InPoint>
                <OutPoint>131600000</OutPoint>
            </MediaFile>
            <MediaFile>
                <GroupType>dest</GroupType>
                <MediaType>FILETYPE_PROXYMP4</MediaType>
                <FileName>\\\\hive.sobey.com\\hivefiles\\sobeyhive\\bucket-p\\streamings\\2016\\07\\18\\videogroup_0.mp4</FileName>
                <InPoint>0</InPoint>
                <OutPoint>131600000</OutPoint>
                <FileParam>
                    <FileFormat>2103</FileFormat>
                    <VideoFormat>37</VideoFormat>
                    <BitRate>4113804</BitRate>
                    <FrameRate>25</FrameRate>
                    <ImageWidth>720</ImageWidth>
                    <ImageHeight>576</ImageHeight>
                    <KeyFrameRate>15</KeyFrameRate>
                    <AudioFormat>1047</AudioFormat>
                    <Channels>1</Channels>
                    <SamplesPerSec>48000</SamplesPerSec>
```

```
                    <BitsPerSample>16</BitsPerSample>
                    <Duration>1941200000</Duration>
                    <FileSize>101872881</FileSize>
                </FileParam>
            </MediaFile>
            <Info>success</Info>
        </RTN_Task>
    </Content>
</MPC>
```

第4节 注意事项

1. 任务类型必须要正确；
2. TaskInfo 不是一定会有的， 如果发送过来的任务里面有， 则需要在任务完成的反馈协议里面返回来；
3. GroupType+MediaType 可以唯一标识一个文件；
4. 任务完成时， 如果有新生成的 MediaFile， 如果 GroupType+MediaType 和源文件一致，那么将替换源文件，如果不一致就进行追加；
5. 接受到任务时，无论成功与否都需要有同步反馈；
6. 如果发送任务是发现任务没有被接受，那么等一段时间后会再次调度，并且这次任务的 JobID 和之前的是一样的；
7. 任务完成时，无论成功与否都需要有异步反馈；
8. AssignTask 协议 JobInfo 的里面 Data 节点；如果是转码，必须要为“ClipMakerJobParam2”，如果是打包，由打包厂商自己定义自己解析。