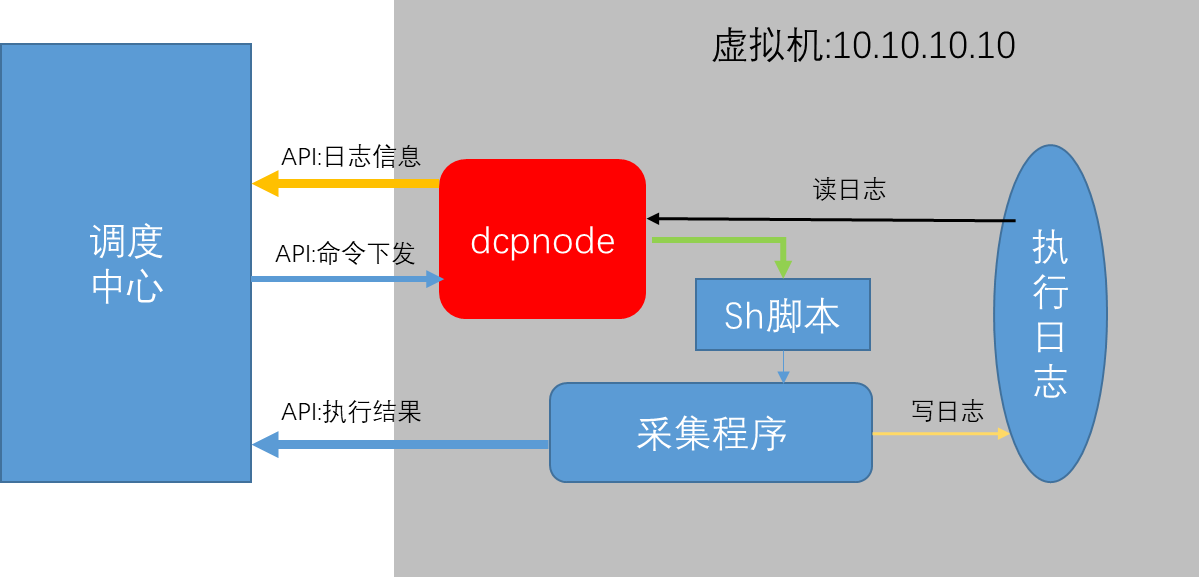
执行器代理程序使用说明

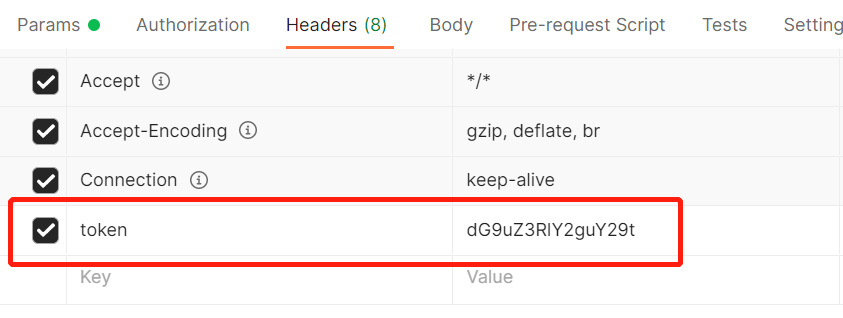
# 接口说明：

为了实现各类采集汇总程序和调度程序解耦，特提供该采集执行器代理程序，该程序通过API方式接收调度程序下发的指令，接收到指令后解析后代理程序所在的主机上的执行命令，命令执行后返回成功信息，等调度的程序完成执行后再次调用调度程序提供的API返回采集汇总程序的实际执行情况信息。



1. API命令下发：

所有API调用执行都必须在Header中带token信息，和dcpnode启动参数一致。



<http://localhost:4321/api/cmd?cmdname=tongsumex.bat&cmdp1=2021-11-11T10:00:00&taskId=12345678>

cmdname：为执行采集汇总脚本，win系统必须bat后缀，linux必须sh后缀

taskId：注意I大写，所有sh脚本中必须把这个作为第1个参数，采集程序将按照这个将日志写入到log/taskId.log

cmdp1:时间参数，可以传两种格式，一种是固定时间格式2021-11-11T10:00:00，中间空格替换为T

另一种格式为NOW|60|50 获取当前时间向前推50分钟后的60分钟整点

cmdp2:其它参数，可以传OMCID等这样的参数

下发执行返回接口参数如下：

{

"taskid":"12345678", //传如的taskId直接返回

"pid":11060, //启动的进程的id

"filelog":"log/12345678.log", //日志路径

"status":true, //调度成功为true，调度失败为false

"errinfo":"" //调度失败后的错误信息

}

1. API执行结果：

程序执行完毕后调用调度中心提供的API返回执行结果，接口参数如下

{

"taskId":"a044a323-67f6-4360-b624-f29996a1033e", //任务ID

"specialty": "WL", //专业类型

"netType": "router", //网元类型

"dataType": "性能", //数据类型

"dataSourceName": "OMC1", //采集源名称

"startTime": "2021-12-20 12:15:00", //业务数据开始时间

"dataCollectGranularity": "15", //数据采集粒度（分钟）

"taskStartTime": "2021-12-20 12:31:23", //任务开始时间

"taskEndTime": "2021-12-20 12:35:45", //任务结束时间

"taskExecutionStatus": "true", //任务状态

"notificationMessageGenerationTime": "2021-12-20 12:35:45" //通知消息生成时间

}

1. API日志信息：

[http://localhost:4321/s/log/{taskId}.log](http://localhost:4321/s/log/%7btaskId%7d.log)

需要采集程序按照传入的参数taskId将日志写入到dcpnode程序所在路径下log目录中，日志命名为taskId+”.log”

# 脚本参考：

五个参数分别对应API中的，taskId，cmdp1，cmdp2，cmdp3，cmdp4

Windows: 使用dcpnode\_win.exe

**tongsumex.bat**

tongsumex -log "%1" -starttime "%2" -omcid "%3" -p3 “%4” -p4 “%5”

**linux：** 使用dcpnode

**tongsumex.sh**

#!/bin/bash

tongsumex -conf cfg/tongsumex.cfg –log “$1” -starttime "$2" -omcid=“$3” -p3 “$4” -p4 “$5”

# dcpnode程序部署和启动：

1. 创建dcpuser用户
2. 创建 /opt/dcp目录，并将所有权给dcpuser
3. 将dcpnode程序放入dcp根目录，并创建log日志路径
4. 将其它相关程序放入/opt/dcp中，注意sh脚本调用路径关系
5. 给dcpnode执行权限
6. 执行dcpnode即可启动

# dcpnode启动参数说明：

-cpulimit float

Denial of service after CPU value exceeds (default 99) 当cpu使用率超过99拒绝接收调度

-memlimit float

Denial of service after MEM value exceeds (default 98) 当内存使用率超过98拒绝接收调度

-p string

dcpnode port (default "4321")

-token string

dcpnode token (default "dG9uZ3RlY2guY29t")

可以启动时通过-p修改监听端口，通过-token修改token鉴权

无

# 数据共享调度执行附加功能：

数据共享时需要下发一次性任务或者定时任务，又调度平台下发任务后执行，为了保证代码的复用性，该程序增加根据前端下发的json串和必要的参数，根据参数配置的模板文件自动进行json字符串查找和拼接为执行程序自动生成配置文件。

前端POST调用http://localhost:4321/api/cmd

{

"spc": {

"dataPeriod": 0,

"extractDetail": {},

"orderNo": "668442471386189824",

"sourceMode": 1,

"workMode": 1,

"workSchedule": "0 0/5 \* \* \* ?",

"columns": [{"name":"aaa"},{"name":"bbb"}],

"targetType": 0,

"targetProperty": {

"schema": "yngx",

"password": "58046b41ca93909f00df1f5758b4b4e1dd5dfe1e58b216297d895529c26916c4",

"port": "3306",

"encryptMode": 1,

"ip": "10.10.90.190",

"dbName": "yngx",

"dbType": "18",

"username": "18f47092fb0b4e77b93bfbc52645e1a622bb8642612bb2c4808996553c63ec2c"

},

"sql": "SELECT \* FROM ca\_cm\_pm\_4g\_port\_q",

"sourceType": 3,

"sourceProperty": {

"schema": "cadb",

"password": "58046b41ca93909f00df1f5758b4b4e1dd5dfe1e58b216297d895529c26916c4",

"encryptMode": 1,

"dbType": 38,

"dbTypeName": "Greenplum",

"url": "jdbc:postgresql://10.10.80.67:5432/SJGXPT",

"username": "18f47092fb0b4e77b93bfbc52645e1a622bb8642612bb2c4808996553c63ec2c"

},

"selector": [],

"startTime": "2021-12-16 10:29:19",

"taskName": "test-db",

"endTime": "2021-12-16 10:29:05",

"extractType": 1,

"targetDetail": {

"loadType": 1,

"operatorType": 0,

"clearData": 1

}

},

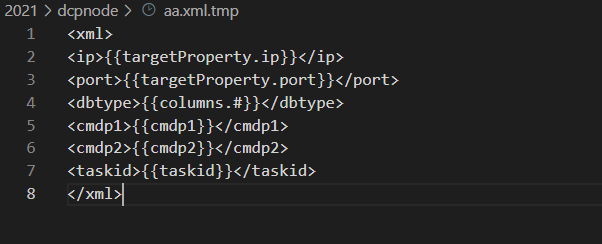
"cmdname": "aa.sh",

"conf": "aa.xml",

"taskId": "49bc4eb3-56a0-4e3b-8c76-d2f4c008dbe4"

}

采集执行器目录下面需要配置aa.xml.tmp如下，需要根据conf参数调整名称



配置的所有参数都需要用{{ }}包裹，语法如下：

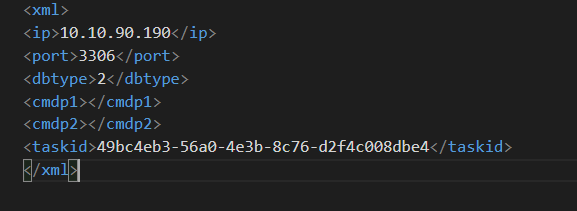
1. 无嵌套参数，比如sql，可以配置为{{ sql }}
2. 有嵌套参数，比如sourceProperty下的ip，可以配置为{{targetProperty.ip}}
3. 其它特殊配置方法

{{columns.#}} 获取数据数量，样例中为2

{{columns.#.name}} 获取数组中指定的字段，然后用|分割，样例中为aaa|bbb

{{columns.0.name}} 获取数组中第一个的name值，样例中为aaa

调用执行后可自动在配置文件目录下自动生成taskid +“.xml”文件，样例中会生成49bc4eb3-56a0-4e3b-8c76-d2f4c008dbe4.xml如下



并执行aa.sh，因此可以再aa.sh配置如下

执行程序时，自动会将配置的taskid作为第一个参数，cmdp1-4作为第二到第五个参数

由于每次都会根据taskid生成配置文件，为了清理配置文件，建议在sh中配置删除xml的逻辑

#!/bin/bash

Java ？？？ -conf $1.xml –log “$1” -starttime "$2" -omcid=“$3” -p3 “$4” -p4 “$5”

rm -rf ./ $1.xml