klustron CDC 使用手册

一．需求背景

klustron分布式数据库为了支持将klustron集群中实时变化数据导入第三方存储系统，便于对数据进行分析查询；或者从第三方开源mysql中实时导入数据到klustron集群中。

二．实现原理

klustron CDC (capture data change)基于mysql binlog dump协议对数据进行实时捕获输出。目前klustron cdc 支持两种模式，一种是从klustron集群导出数据，一种是支持开源mysql实例导出数据。

* klustron集群导出

klustron CDC根据dump 任务参数，连接到klustron元数据集群，获取dump cluster中shard参数，如果有多个shard，CDC会自动为每个shard建立dump连接。选择shard节点中延迟最小的备节点进行dump数据。如果dump过程中shard发生主备切换，即当前dump节点为主节点或者dump节点挂掉等情况。CDC自动断开当前dump连接，重新选择shard中其他延迟最小的备节点进行dump。

* 开源mysql导出

klustron CDC根据dump任务参数，连接到需要dump的mysql节点。目前针对开源mysql，CDC会实时检测dump连接情况，如果dump过程中连接断开，例如网络问题，源mysq中kill连接等情况，CDC自动重新连接。

klustron CDC采用插件方式将捕获数据输出到目标系统。CDC提供开发插件API接口，用户可根据CDC 提供API接口开发自定义目标存储。目前CDC发布时自带两个插件event\_file和event\_sql。klustron CDC将捕获数据同步目标存储时，能够保证数据不丢失。CDC会根据dump数据gtid信息，实时备份同步点。如果dump过程中CDC模块重启，启动后自动根据上次同步点继续同步。CDC模块支持集群部署，实现高可用。

三．配置使用

1.通过API方式配置

1.1 添加dump数据任务

1.1.1 从klustron集群dump数据

1> 指定从具体位置开始dump

**curl -d '**

{  
    **"version"**:**"1.0"**,  
    **"job\_id"**:**""**,  
    **"job\_type"**:**"add\_dump\_table"**,  
    **"timestamp"**:**"1435749309"**,  
    **"user\_name"**:**"kunlun\_test"**,  
    **"paras"**:{  
        **"meta\_db"**:**"****172.0.0.1:28001,172.0.0.2:28001,172.0.0.3:28001"**,  
        **"meta\_user"**:**"xxx"**,  
        **"meta\_passwd"**:**"xxxx"**,  
        **"cluster\_name"**:**"cluster\_xxx\_xxx"**,  
        **"dump\_tables"**:**"postgres\_$$\_public.t1,postgres\_$$\_public. t2"**,  
        **"shard\_params"**:[  
            {  
                **"shard\_id"**:**"1"**,  
                **"dump\_hostaddr"**:**"127.0.0.1"**,  
                **"dump\_port"**:**"28801"**,  
                **"binlog\_file"**:**"xxx"**,  
                **"binlog\_pos"**:**"899"**,  
                **"gtid\_set"**:**"xxxx"**  
            },{  
                **"shard\_id"**:**"2"**,  
                **"dump\_hostaddr"**:**"127.0.0.2"**,  
                **"dump\_port"**:**"28802"**,  
                **"binlog\_file"**:**"xxx"**,  
                **"binlog\_pos"**:**"899"**,  
                **"gtid\_set"**:**"xxxx"**  
            }  
        ],  
        **"output\_plugins"**:[  
            {  
                **"plugin\_name"**:**"event\_file"**,  
                **"plugin\_param"**:**"/xxx/event.log"**,  
                **"udf\_name"**:**"test1"**  
            },  
            {  
                **"plugin\_name"**:**"event\_sql"**,  
                **"plugin\_param"**:**"{\"hostaddr\":\"172.0.0.5\",\"port\":\"24002\",\"user\":\"xxxx\",\"password\":\"xxx\",\"log\_path\":\"../log\"}"**,  
                **"udf\_name"**:**"test2"**  
            }  
        ]  
    }  
}**' -X POST** [**http://172.0.0.1:18002/kunlun\_cdc**](http://172.0.0.1:18002/kunlun_cdc)

2> 从当前添加dump任务时间点开始dump

**curl -d '**

**{**

**"version":"1.0",**

**"job\_id":"",**

**"job\_type":"add\_dump\_table",**

**"timestamp":"1435749309",**

**"user\_name":"kunlun\_test",**

**"paras":{**

**"meta\_db":"172.0.0.1:28001,172.0.0.2:28001,127.0.0.3:28001",**

**"meta\_user":"xxx",**

**"meta\_passwd":"xxx",**

**"cluster\_name":"cluster\_xxx\_xx",**

**"dump\_tables":"postgres\_$$\_public.t1,postgres\_$$\_public.t2",**

**"output\_plugins":[{**

**"plugin\_name":"event\_file",**

**"plugin\_param":"/home/barney/kunlun\_cdc/temp/event.log",**

**"udf\_name":"test1"**

**},{**

**"plugin\_name":"event\_sql",**

**"plugin\_param":"{\"hostaddr\":\"172.0.0.6\",\"port\":\"24002\",\"user\":\"xxx\",\"password\":\"xxx\",\"log\_path\":\"../log\"}",**

**"udf\_name":"test2"**

**}]**

**}**

**}**

**' -X POST** [**http://172.0.0.1:18002/kunlun\_cdc**](http://172.0.0.1:18002/kunlun_cdc)

1.1.2 从开源mysql集群dump数据，必须指定dump具体位置，即设置shard\_params参数

**curl -d '**

{  
    **"version"**:**"1.0"**,  
    **"job\_id"**:**""**,  
    **"job\_type"**:**"add\_dump\_table"**,  
    **"timestamp"**:**"1435749309"**,  
    **"user\_name"**:**"kunlun\_test"**,  
    **"paras"**:{  
        **"meta\_db"**:**"127.0.0.1:28001"**, --- dump mysql的ip:port  
        **"meta\_user"**:**"xxx"**, --- 连接mysql的账户  
        **"meta\_passwd"**:**"xxx"**, --- 连接mysql的密码  
        **"cluster\_name"**:**"mysql"**,  
        **"dump\_tables"**:**"test.t1,test.t2"**,  
        **"is\_kunlun"**:**"0"**,  
        **"shard\_params"**:[{  
            **"binlog\_file"**:**"xxx"**,  
            **"binlog\_pos"**:**"899"**,  
            **"gtid\_set"**:**"xxxx"**  
        }],  
        **"output\_plugins"**:[  
            {  
                **"plugin\_name"**:**"event\_file"**,  
                **"plugin\_param"**:**"/xx/event.log"**,  
                **"udf\_name"**:**"test1"**  
            },  
            {  
                **"plugin\_name"**:**"event\_sql"**,  
                **"plugin\_param"**:**"{\"hostaddr\":\"172.0.0.2\",\"port\":\"24002\",\"user\":\"abc\",\"password\":\"abc\",\"log\_path\":\"../log\"}"**,  
                **"udf\_name"**:**"test2"**  
            }  
        ]  
    }  
} **' -X POST** [**http://172.0.0.1:18002/kunlun\_cdc**](http://172.0.0.1:18002/kunlun_cdc)

1.2 删除dump 数据任务

curl -d '

{

"version":"1.0",

"job\_id":"",

"job\_type":"del\_dump\_table",

"timestamp":"1435749309",

"user\_name":"kunlun\_test",

"paras":{

"meta\_db":"172.0.0.1:28001,172.0.0.2:28001,172.0.0.3:28001",

"cluster\_name":"cluster\_xxxx\_xxx",

"dump\_tables":"postgres\_$$\_public.t1,postgres\_$$\_public.t2"

}

}

' -X POST <http://172.0.0.1:18002/kunlun_cdc>

1.3 获取当前CDC集群主节点

curl -d '

{

"version":"1.0",

"job\_id":"",

"job\_type":"get\_leader",

"timestamp":"1435749309",

"user\_name":"kunlun\_test"

}

' -X POST <http://172.0.0.1:18002/kunlun_cdc>

1.4 获取当前CDC集群支持的同步目标插件

curl -d '

{

"version":"1.0",

"job\_id":"",

"job\_type":"list\_support\_plugins",

"timestamp":"1435749309",

"user\_name":"kunlun\_test"

}

' -X POST <http://172.0.0.1:18002/kunlun_cdc>

1.5 获取当前CDC集群中所有dump任务

curl -d '

{

"version":"1.0",

"job\_id":"",

"job\_type":"list\_dump\_jobs",

"timestamp":"1435749309",

"user\_name":"kunlun\_test"

}

' -X POST <http://172.0.0.1:18002/kunlun_cdc>

1.6 获取某个dump任务同步状态

curl -d '

{

"version":"1.0",

"job\_id":"",

"job\_type":"get\_job\_state",

"timestamp":"1435749309",

"user\_name":"kunlun\_test",

"paras":{

"meta\_db":"172.0.0.1:28001,172.0.0.2:28001,172.0.0.3:28001",

"cluster\_name":"cluster\_xxx\_xxx",

"dump\_tables":"postgres\_$$\_public.t1,postgres\_$$\_public.t2"

}

}

' -X POST http://172.0.0.1:18002/kunlun\_cdc

2.通过xpanel方式配置

四．输出json格式以及支持sql语句

输出json格式

|  |  |
| --- | --- |
| 字段名 | 说明 |
| database | 数据库名 |
| table | 表名 |
| isDdl | 是否为ddl |
| sql | Ddl 执行sql语句 |
| event\_type | 事件名称 |
| data | 如果为insert，为插入每列对于数据  如果为delete，为删除数据，  如果为update，为更新后行 |
| old | 如果为insert，空  如果为delete，空，  如果为update，更新前数据 |

目前支持event type

|  |  |
| --- | --- |
| 字段名 | 说明 |
| CREATE\_DB | 创建库 |
| DROP\_DB | 删除库 |
| CREATE\_TABLE | 创建表 |
| DROP\_TABLE | 删除表，该语句支持多个表同时进行，kunlun\_cdc这个地方将多个表拆成多个drop table 记录 |
| CREATE\_INDEX | 添加索引 |
| DROP\_INDEX | 删除索引 |
| ALTER\_TABLE | 表添加，删除和更新字段等 |
| RENAME\_TABLE | 表重命名，该语句支持多个表同时进行，kunlun\_cdc这个地方将多个表拆成多个rename table 记录 |
| INSERT | 插入数据 |
| DELETE | 删除数据 |
| UPDATE | 更新数据 |

五．使用klustron cdc 前置条件

dump的源db 需要配置

1. gtid\_mode=ON，否则无法保证数据不丢失

2. binlog\_row\_metadata=FULL，否则CDC无法正常工作

3. binlog\_row\_image=FULL，建议设置为FULL。

六．自带插件说明：

event\_file插件直接将CDC捕获的数据转成json写入文件。开发自定义插件时可以参考event\_file插件输出。

event\_sql支持将CDC捕获的数据转换成sql，写入目标db。目标db可以为klustron集群或者开源mysql。默认写入klustron集群，如果需要开源mysql需要在plugin\_param中添加is\_kunlun=0。

七．klustron CDC部署说明

1.获取kunlun\_cdc安装包，解压到目标目录下。

2. 到conf目录下修改kunlun\_cdc.cnf文件

# Copyright (c) 2022 ZettaDB inc. All rights reserved.

# This source code is licensed under Apache 2.0 License,

# combined with Common Clause Condition 1.0, as detailed in the NOTICE file.

[Base\_Config]

############################################

# base config

local\_ip = 172.0.0.1

http\_port = 18002

log\_file\_path = ../log/kunlun\_cdc

log\_file\_size = 500

基础配置

Local\_ip为本机ip地址

http\_port为klustron CDC监听到端口

log\_file\_path和log\_file\_size 为kunlun\_cdc日志配 置

[Binlog\_Config]

############################################

# connect cluster shards strategy

allow\_dump\_shard\_master = 0

dump\_shard\_node\_max\_delay = 1800

loop\_query\_shard\_state = 10

loop\_report\_cdc\_sync\_state = 5

binlog\_msg\_queue\_len = 1024

pending\_binlog\_event\_num = 1000

reserve\_binlog\_event\_dir = ../data/reserve\_dir

kunlun\_cdc dump binlog相关配置

allow\_dump\_shard\_master是否允许从shard主上dump，默认为0

dump\_shard\_node\_max\_delay 为dump的shard 备节点允许的最大延迟，如果该节点延迟大约配置值，则kunlun\_cdc自动选择shard其他备节点

loop\_query\_shard\_state 查询dump shard 状态时间间隔

loop\_report\_cdc\_sync\_state dump表状态固化时间间隔

[HA\_Config]

############################################

# config paxos

ha\_group\_member = 172.0.0.1:18001,172.0.0.2:18001,172.0.0.3:18001

server\_id = 2

paxosdata\_dir = ../data/paxosdata

paxoslog\_dir = ../data/paxoslog

paxosdata\_compress = 0

paxosdata\_write\_buffer\_size = 2

paxosdata\_max\_write\_buffer\_number = 2

paxosdata\_min\_writer\_buffer\_number\_to\_merge = 1

paxosdata\_max\_backgroup\_compactions = 6

paxosdata\_max\_bytes\_for\_level\_base = 64

paxosdata\_target\_File\_size\_base = 64

paxosdata\_level0\_slowdown\_writes\_trigger = 12

paxosdata\_level0\_stop\_writes\_trigger = 16

paxosdata\_block\_cache\_size = 5

paxosdata\_block\_size = 64

paxosdata\_bloom\_filter\_bits\_per\_key = 10

paxosdata\_block\_based\_bloom\_filter = 0

klustron CDC高可用配置

ha\_group\_member klustron CDC集群节点ip:port，个数要求为奇数

server\_id为该节点ip在ha\_group\_member中位置，例如该机器为172.0.0.2，则server\_id=2

[Plugin\_Config]

############################################

plugin\_so = event\_file,event\_sql

klustron CDC插件配置。

用户开发自定义插件，需要追加到这个地方。重启CDC服务才能生效。