# klustron CDC 使用手册

## 一．需求背景

Klustron CDC 用于将klustron分布式数据库集群中的各个存储集群的数据更新实时导出为事件流，供外部组件消费。例如，外部组件可以做数据流式导入，把klustron的数据更新导入到第三方数据库系统，或者导出到flinks等事件流分析系统；Klustron CDC还支持从第三方开源mysql中实时导出数据更新事件流，这样就可以流式导入数据到klustron集群中。

## 二．实现原理

klustron CDC (change data capture )基于mysql binlog dump协议对数据进行实时捕获输出，对mysql实例来说，Klustron CDC组件就像一个备机节点。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，获取需要dump 数据的klustron cluster中shard参数，如果有多个shard，CDC会自动为每个shard建立dump连接。klustron CDC会选择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会自动断开当前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mysql中kill连接等情况，CDC自动重新连接。

### 事件输出和转换插件

klustron CDC采用插件方式将捕获的binlog 数据做转换和输出。Klustron 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，挂载到Klustron CDC。目前Klustron CDC发布时自带两个插件event\_file和event\_sql。

Event\_file的功能是：直接将CDC输出json内容序列化存储到指定文件。

Event\_sql的功能是：将CDC输出json内容转换成sql语句，直接发送给klustron集群或者开源mysql

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模块因为各种软硬件故障而退出，那么重新启动后能够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。

如果需要实现其他转换插件，则需要按照下述插件接口实现转换插件，然后挂载到klustron CDC。

### 事件输出接口说明

CDC输出json格式

|  |  |
| --- | --- |
| 字段名 | 说明 |
| gtid | 当前事件gtid |
| 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 |

支持输出事件类型

|  |  |
| --- | --- |
| 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 |

对于DDL语句，json字段isDdl=1，sql记录当前ddl语句，例如：

{"event\_type":"CREATE\_TABLE","db\_name":"test","sql":"create table t (a int primary key, b int)","isDdl":"1","table\_name":"t","data":"","old":"","gtid":"77cf0403-fe85-11ed-87ad-fc3497a73395:5620"}

对于DML语句，json字段isDdl=0， 在data和old字段中记录具体内容。例如：

{"event\_type":"INSERT","db\_name":"test","table\_name":"t","sql":"","isDdl":"0","gtid":"77cf0403-fe85-11ed-87ad-fc3497a73395:5621","data":[{"a":"1","b":"1"}], "old":""}

### 自定义插件开发以及挂载方法

1.需要去klustron官网下载开发插件依赖的头文件dispatch\_event.h

2. 继承CDispatchEvent类，实现init/execute/close三个虚函数接口，在dispatch\_event.h中有详细说明各个函数使用方法

3. 编译成so，放到kunlun\_cdc安装包的plugin目录下。

4. 在kunlun\_cdc安装包的conf目录下kunlun\_cdc.cnf中

plugin\_so标签下添加开发插件名。

例如开发插件so名为event\_test.so。

在kunlun\_cdc.cnf文件plugin\_so标签下添加内容如下

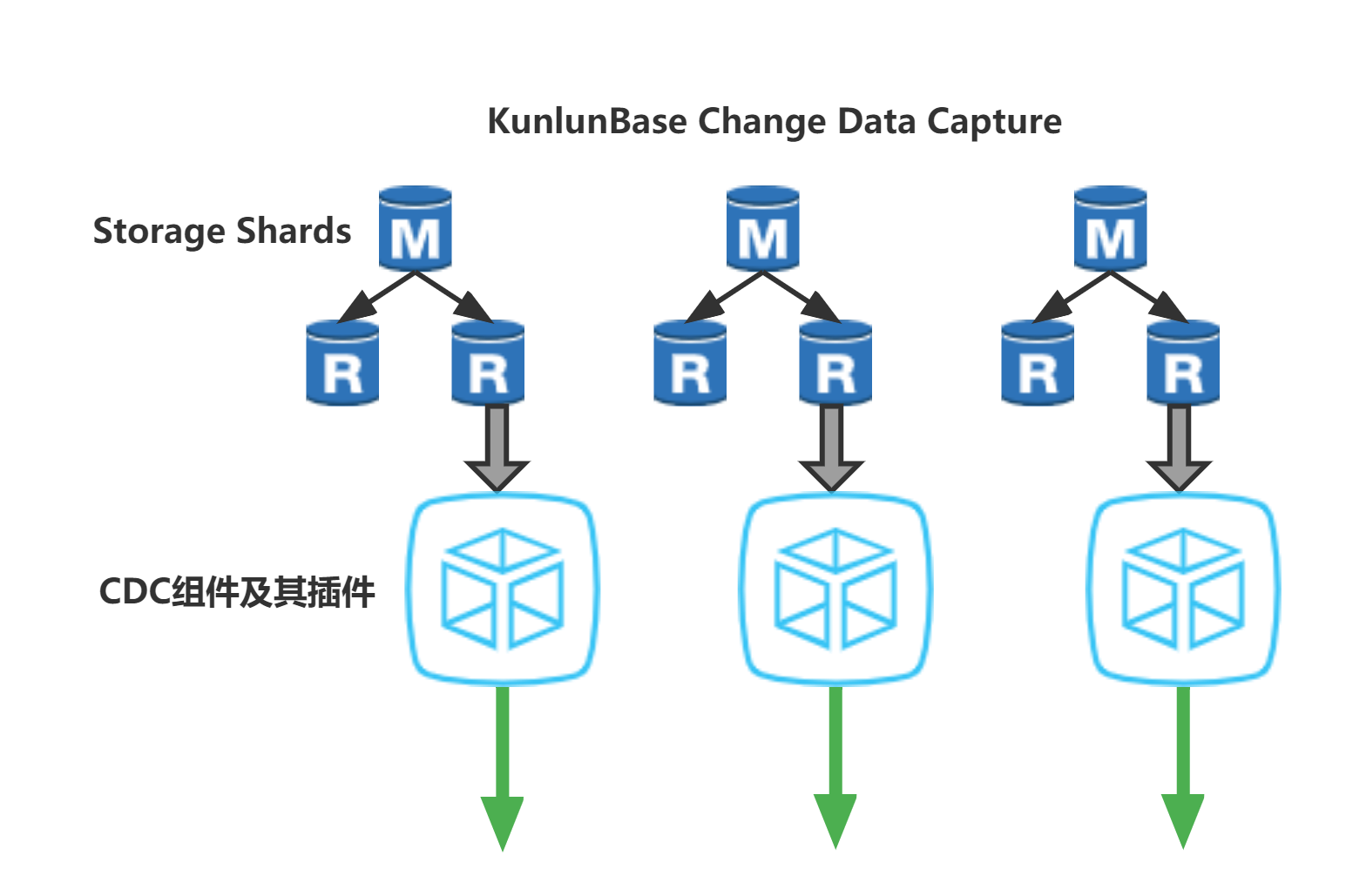
plugin\_so = event\_file,event\_sql,event\_test

特别注意：插件名之间用逗号隔开

5. 修改完配置文件，需要重启klustron CDC集群所有CDC进程才可以生效

由于插件用户自己开发，那插件需要输入参数，则在添加dump任务时output\_plugins标签下输入该插件需要参数。

## 三．部署架构图



## 四．配置使用

1.通过API方式配置

1.1 添加dump数据任务（异步接口）

1.1.1 从klustron集群dump数据

1> 指定从具体位置开始dump，需要指定该cluster下各个shard开始dump具体位置，必须指定binlog\_file，binlog\_pos以及gtid\_set。

**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，如果不指定则添加任务失败。

**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（异步接口）

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（同步接口）

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（同步接口）

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（同步接口）

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 （同步接口）

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>

1.7 获取异步任务状态接口（同步接口）

curl -d '

{

"version":"1.0",

"job\_id":"xxx", --需要查询任务job\_id

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

"timestamp":"1435749309",

"user\_name":"kunlun\_test"

}

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

1.8 获取CDC集群配置信息接口（同步接口）

curl -d '

{

"version":"1.0",

"job\_id":"",

"job\_type":"list\_cdc\_conf",

"timestamp":"1435749309",

"user\_name":"kunlun\_test"

}

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

2.通过xpanel方式配置

2.1 上报CDC集群到xpanel

Graphical user interface, application

Description automatically generated

点击CDC服务，新增按钮

Graphical user interface, timeline

Description automatically generated

上报CDC服务成功。

2.2 添加CDC任务

Graphical user interface, table

Description automatically generated

点击CDC任务，新增按钮

设置dump任务数据源，从开源mysql导出。

Graphical user interface, application

Description automatically generated

设置同步点数据信息

A picture containing background pattern

Description automatically generated

点击确认保存

设置数据同步到目标源，可以配置多个目标源

1. 配置数据json文件

Icon

Description automatically generated with low confidence

点击确认保存

1. 配置同步到kunlunbase

A picture containing application

Description automatically generated

点击确认保存

Graphical user interface, application

Description automatically generated

点击确认下发任务

Text

Description automatically generated

2.3 删除CDC任务

从xpanel CDC任务页面找到对应业务

Graphical user interface

Description automatically generated

点击删除，输入验证码确认则发起删除任务

Graphical user interface, text, application

Description automatically generated

2.4 查看CDC同步状态

从xpanel CDC任务页面找到对应业务

Graphical user interface, text, application

Description automatically generated

点击详情可以查看具体同步情况

Graphical user interface, application, table

Description automatically generated

## 五．使用klustron cdc 前置条件

dump的源mysql db 需要以下配置

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：

6.1 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\_file 插件输入参数为json内容具体写入的文件。添加dump任务时output\_plugins字段中输入参数，例如

**"plugin\_name"**:**"event\_file"**, -- 插件名称  
**"plugin\_param"**:**"/xx/event.log"**, -- 表示json内容写入该event.log文件  
**"udf\_name"**:**"test1" -- 扩展字段**

6.2 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。

event\_sql插件输入参数，添加dump任务时output\_plugins字段中输入参数，例如

## "plugin\_name":"event\_sql", --- 插件名称 "plugin\_param":"{\"hostaddr\":\"172.0.0.2\",\"port\":\"24002\",\"user\":\"abc\",\"password\":\"abc\",\"log\_path\":\"../log\"}",---event\_sql插件输入参数 "udf\_name":"test2" --- 扩容字段

特别说明：如果使用event\_sql将源数据写入klustron节点，plugin\_param里面配置为klustron计算节点mysql 端口，不能为pg端口。

## 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，该配置在从kunlunbase集群dump数据时有效

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。该配置在从kunlunbase集群dump数据时有效

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

pending\_binlog\_event\_num 表示在xa事务中，CDC默认可以缓存多少条binlog event，当大于配置值时，CDC将缓存binlog event消息写入磁盘

reserve\_binlog\_event\_dir 表示CDC可以将缓存binlog event消息写入磁盘位置

[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。