启动instance

心跳

心跳

心跳

NM

NM

NM

CanalCenter-SLAVE

CanalCenter-Master

……

NodeN

Node2

Node1

运行instance的Node的集群

ZK集群

1：CanalCenter由多台机器组成,通过分布式锁控制 只有一台作为Master,其他的SLAVE作为”冷备份”,当Master挂掉以后,SLAVE抢到分布式锁以后,执行Master职责,CanalCenter启动成功以后需要向ZK注册,这样Node就可以知道Server的机器

2:每个Node上面有一个NodeManager进程,该进程主要有两个职责

2.1:向CanalCenter发送心跳,表名自己是一个活跃的结点

2.2:负责处理CanalCenter给他发送的Command(一般为启动instance,类似于MR中的Task)

HA设计

1：当instance挂掉以后,基于zookeeper的listener机制,可以监控到instance下线,CanalCenter会在其他的node上面启动instance继续处理

2:当Node挂掉以后,同样 基于zookeeper的listener机制,将这个Node上面的所有instance(Zookeeper上面获取)先全部停止(更新Zookeeper结点上的状态,instance运行期间会判断该状态,如果为STOP就停在处理数据,然后sleep一段时间,等待binlog的position同步到zookeeper中,然后自己退出),退出以后Zookeeper检测到会在其他机器启动该instance

ZK的树形结构

橙色表示”瞬时”结点,生命周期和创建该结点的进程不同,进程挂了,该结点就消失了

绿色结点为”持久”结点,除非手动删除,否则会一直都在

/datacanal/lock/ 该目录是分布式锁目录

/datacanal/lock/serverlock/ 这是canal server启动用到的分布式锁的

/datacanal/canal\_server/{server addr} 这边是存储服务端的IP和端口

/datacanal/node/{noden} 这边是存储所有的”运行结点的信息”

/datacanal/task 这边是存储所有的抽取任务

/datacanal/task/logictable 任务的逻辑表

/datacanal/task/logictable/physicstable 逻辑表的分片信息(物理表)

/datacanal/task/logictable/physicstable/instance/{instance}

canal server控制instance的运行

/datacanal/task/logictable/physicstable/position 该分片消费到的位置

Q&A

1：Parser处理的slave挂了如何处理？

当parser处理的slave挂了,需要切换到同样的master的其他slave下面,这里存在如下两个问题

1.1：调度系统需要知道master和slave之间的关系,这样才能做到”切换到其他的slave上面”

1.2：切换slave后binlog的定位 ,对于不同的slave的binlog和position是不统一的,但是总体是一致的

比如：slave1的mysql-00001.log

关于切换slave以后binlog的定位

当slave1挂掉以后,Parser会重试几次,重试超过一定的额阈值以后认为该结点已经挂了,然后记录下

position,然后寻找同master的slave2,从当前的position向前寻找,直到找到相同的Message

注意点：

1：slave-001挂掉后,parser会重试一段时间(3分钟), slave-002的binlog一定已经领先了slave-001了

2：slave-002最多往前找10分钟(slave之间的延迟是秒级的,所以找10分钟一定能找到),

2:如何启动datacanal-center

com.canal.center. CanalCenterLauncher.java

3:如何启动datacanal-instance-manager

com.canal.instance.manager.code. NodeManagerLaucher.java

4: datacanal-instance被集成到了datacanal-instance-manager

datacanal-instance的启动由manager中调用启动脚本启动

(datacanal-instance-manager/resource/start.sh)