FlinkCDC聚合mysql及pg异构数据源并宽表压入es实验

1. 环境搭建
   1. pg

略

* 1. es+kibana

略

* 1. mysql

略

* 1. hadoop集群

略

* 1. flink1.13集群
     1. 下载flink1.13
     2. 配置

flink-conf.yaml：

jobmanager.rpc.address: node1

taskmanager.numberOfTaskSlots: 2

parallelism.default: 1

# 高可用配置

state.backend: filesystem

state.backend.fs.checkpointdir: hdfs://node1:9000/flink-checkpoints

high-availability: zookeeper

high-availability.storageDir: hdfs://node1:9000/flink/ha/

high-availability.zookeeper.quorum: node1:2181,node2:2181,node3:2181,node4:2181

high-availability.zookeeper.client.acl: open

web.submit.enable: true

masters:

node1:8081

node2:8081

workers:

node1

node2

node3

node4

* 1. flinkCDC2.1connector

把用到的 flinkCDC 三个文件下载

flink-sql-connector-elasticsearch7\_2.11-1.13.2.jar

flink-sql-connector-mysql-cdc-2.1.0.jar

flink-sql-connector-postgres-cdc-2.1.0.jar

放入flink的lib里

然后把/opt/flink-1.13.3/ 纷发给其他节点上

在节点node2上修改flink-conf.yaml 文件，jobmanager.rpc.address: node2

1. 数据准备
   1. mysql

-- MySQL

CREATE DATABASE mydb;

USE mydb;

CREATE TABLE products (

id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

description VARCHAR(512)

);

ALTER TABLE products AUTO\_INCREMENT = 101;

INSERT INTO products

VALUES (default,"scooter","Small 2-wheel scooter"),

(default,"car battery","12V car battery"),

(default,"12-pack drill bits","12-pack of drill bits with sizes ranging from #40 to #3"),

(default,"hammer","12oz carpenter's hammer"),

(default,"hammer","14oz carpenter's hammer"),

(default,"hammer","16oz carpenter's hammer"),

(default,"rocks","box of assorted rocks"),

(default,"jacket","water resistent black wind breaker"),

(default,"spare tire","24 inch spare tire");

CREATE TABLE orders (

order\_id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

order\_date DATETIME NOT NULL,

customer\_name VARCHAR(255) NOT NULL,

price DECIMAL(10, 5) NOT NULL,

product\_id INTEGER NOT NULL,

order\_status BOOLEAN NOT NULL -- Whether order has been placed

) AUTO\_INCREMENT = 10001;

INSERT INTO orders

VALUES (default, '2020-07-30 10:08:22', 'Jark', 50.50, 102, false),

(default, '2020-07-30 10:11:09', 'Sally', 15.00, 105, false),

(default, '2020-07-30 12:00:30', 'Edward', 25.25, 106, false);

* 1. pg

-- PG

CREATE TABLE shipments (

shipment\_id SERIAL NOT NULL PRIMARY KEY,

order\_id SERIAL NOT NULL,

origin VARCHAR(255) NOT NULL,

destination VARCHAR(255) NOT NULL,

is\_arrived BOOLEAN NOT NULL

);

ALTER SEQUENCE public.shipments\_shipment\_id\_seq RESTART WITH 1001;

ALTER TABLE public.shipments REPLICA IDENTITY FULL;

INSERT INTO shipments

VALUES (default,10001,'Beijing','Shanghai',false),

(default,10002,'Hangzhou','Shanghai',false),

(default,10003,'Shanghai','Hangzhou',false);

* 1. flink临时表

在flink主节点 启动集群

bin/start-cluster.sh

然后跑到yarn的master 节点 node2 上 启 sql-clients

bin/sql-client.sh

顺序执行：

Flink SQL> SET execution.checkpointing.interval = 3s;

CREATE TABLE orders (

order\_id INT,

order\_date TIMESTAMP(0),

customer\_name STRING,

price DECIMAL(10, 5),

product\_id INT,

order\_status BOOLEAN,

PRIMARY KEY (order\_id) NOT ENFORCED

) WITH (

'connector' = 'mysql-cdc',

'hostname' = '192.168.2.180',

'port' = '3308',

'username' = 'root',

'password' = 'Zkr123',

'database-name' = 'mydb',

'table-name' = 'orders'

);

CREATE TABLE products (

id INT,

name STRING,

description STRING,

PRIMARY KEY (id) NOT ENFORCED

) WITH (

'connector' = 'mysql-cdc',

'hostname' = '192.168.2.180',

'port' = '3308',

'username' = 'root',

'password' = 'Zkr123',

'database-name' = 'mydb',

'table-name' = 'products'

);

CREATE TABLE shipments (

shipment\_id INT,

order\_id INT,

origin STRING,

destination STRING,

is\_arrived BOOLEAN,

PRIMARY KEY (shipment\_id) NOT ENFORCED

) WITH (

'connector' = 'postgres-cdc',

'hostname' = '192.168.2.180',

'port' = '5432',

'username' = 'postgres',

'password' = 'postgres',

'database-name' = 'postgres',

'schema-name' = 'public',

'table-name' = 'shipments'

);

CREATE TABLE enriched\_orders (

order\_id INT,

order\_date TIMESTAMP(0),

customer\_name STRING,

price DECIMAL(10, 5),

product\_id INT,

order\_status BOOLEAN,

product\_name STRING,

product\_description STRING,

shipment\_id INT,

origin STRING,

destination STRING,

is\_arrived BOOLEAN,

PRIMARY KEY (order\_id) NOT ENFORCED

) WITH (

'connector' = 'elasticsearch-7',

'hosts' = 'http://192.168.2.180:9200',

'index' = 'enriched\_orders'

);

下面join到一起往es中压

-- Flink SQL

Flink SQL> INSERT INTO enriched\_orders

SELECT o.\*, p.name, p.description, s.shipment\_id, s.origin, s.destination, s.is\_arrived

FROM orders AS o

LEFT JOIN products AS p ON o.product\_id = p.id

LEFT JOIN shipments AS s ON o.order\_id = s.order\_id;

1. 运行看测试结果
   1. es看宽表数据

http://bigdata:5601/ 这是发布的kibana地址

创建 index pattern enriched\_orders

然后就可以在discover里看到数据了，insert的宽表数据

* 1. 数据变动后看es数据

增加和删除后 去看看es数据及时同步过来没有

增加

--PG

INSERT INTO shipments

VALUES (default,10004,'Shanghai','Beijing',false);

在 MySQL 的 orders 表中更新订单的状态

--MySQL

UPDATE orders SET order\_status = true WHERE order\_id = 10004;

在 Postgres 的 shipment 表中更新物流的状态

----------------------------------------

去看es数据增加了一行

--PG

UPDATE shipments SET is\_arrived = true WHERE shipment\_id = 1004;

去看es数据1004 那行数据状态变化了

我发现必须用kibana看数据，以前用的chrome + head插件不成，数据延迟不显示变化状态，难道是浏览器缓存过去结果？

----------------------------------------

删除

--MySQL

DELETE FROM orders WHERE order\_id = 10004;

这是es数据少了一行

1. 总结

有flinkCDC ，直接把汇聚和 聚合压宽表，或者 group 统计查询 在一个flink job里都做了，更简洁直接了

用flinkSQL 直接写sql 方式 告诉job从哪抽数据，怎么聚合怎么transform 然后存哪，不用写代码。