Flink+TiDB

一、FlinkSQL+TiDB实现物化视图

在尝试各个版本的坑之后，最终选择Flink 1.13.6版本。组最简单的搭建方式。

TiDB集群中控机：172.16.11.70

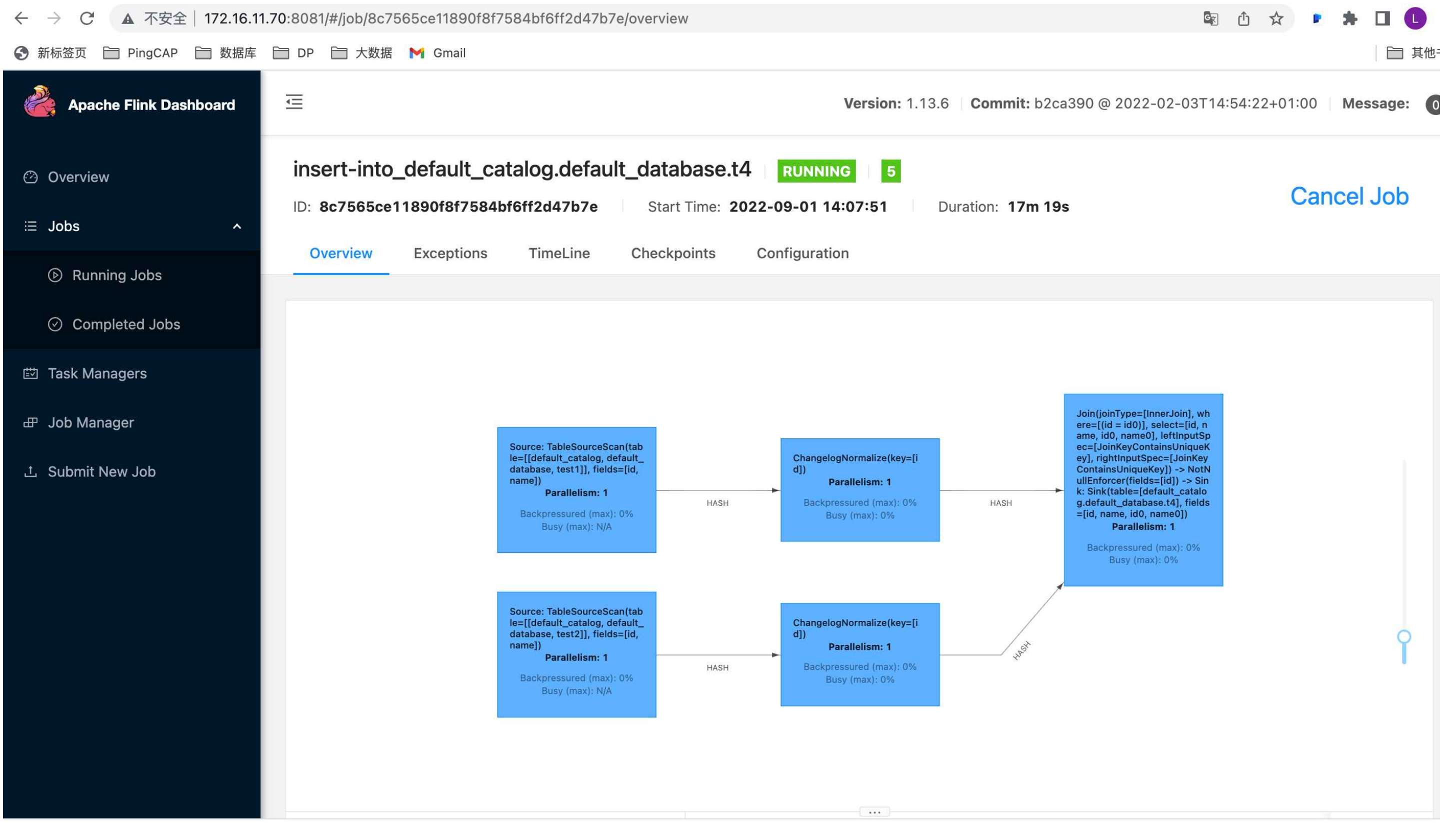
```
[root@ogg bin]# hostname -I
172.16.11.70 172.17.0.1
[root@ogg bin]#
1.版本选择
[root@ogg bin]# ./flink -v
Version: 1.13.6, Commit ID: b2ca390
2.安装
tar xvf flink-1.13.6-bin-scala_2.11.tgz
cd flink-1.13.6/conf/
# 这里单机启动，ip等未做修改。
vim flink-conf.yaml
# 该参数如果是1，复杂任务无法完成
taskmanager.numberOfTaskSlots: 1 ---> 40
# 相应增加内存
jobmanager.memory.process.size: 16000m
taskmanager.memory.process.size: 17280m
3.在/home/ogg/flink-1.13.6/lib添加所需依赖的对应版本jar包
flink-connector-jdbc_2.11-1.13.6.jar
flink-sql-connector-tidb-cdc-2.2.0.jar
mysql-connector-java-8.0.30.jar
4.启动
[root@ogg bin]# pwd
/home/ogg/flink-1.13.6/bin
[root@ogg bin]# ./start-cluster.sh
5.起flink sql
[root@ogg bin]# pwd
/home/ogg/flink-1.13.6/bin
[root@ogg bin]# ./sql-client.sh
6.分别在tidb中创建test1，test2，users表
mysql> show create table test1 \G;
*************************** 1. row ***************************
       Table: test1
Create Table: CREATE TABLE `test1` (
  `id` int(11) NOT NULL,
  `name` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`id`) /*T![clustered_index] NONCLUSTERED */
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin
1 row in set (0.00 sec)

mysql> show create table test2 \G;
*************************** 1. row ***************************
       Table: test2
Create Table: CREATE TABLE `test2` (
  `id` int(11) NOT NULL,
  `name` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`id`) /*T![clustered_index] NONCLUSTERED */
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin
1 row in set (0.00 sec)

mysql> show create table users \G;
*************************** 1. row ***************************
       Table: users
Create Table: CREATE TABLE `users` (
  `id` int(11) NOT NULL,
  `name` varchar(50) DEFAULT NULL,
  `id0` int(11) NOT NULL DEFAULT '1',
  `name0` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`id`) /*T![clustered_index] CLUSTERED */
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin
1 row in set (0.00 sec)

7.分别在flink sql中创建test1, test2, users表对应的映射表test1, test2, t4
Flink SQL> SET 'execution.checkpointing.interval' = '1s';
Flink SQL> CREATE TABLE test1 (
>   id INT,
>   name STRING,
>   PRIMARY KEY(id) NOT ENFORCED
> ) with (
>      'connector' = 'tidb-cdc',
>      'tikv.grpc.timeout_in_ms' = '20000',
>      'pd-addresses' = '172.16.11.44:2381',
>      'database-name' = 'oggadmin',
>      'table-name' = 'test1'
> );
Flink SQL> CREATE TABLE test2 (
>   id INT,
>   name STRING,
>   PRIMARY KEY(id) NOT ENFORCED
> ) with (
>      'connector' = 'tidb-cdc',
>      'tikv.grpc.timeout_in_ms' = '20000',
>      'pd-addresses' = '172.16.11.44:2381',
>       'database-name' = 'oggadmin',
>       'table-name' = 'test2'
> );

Flink SQL> CREATE TABLE t4 (
>    id INT,
>    name STRING,
>    id0 INT,
>    name0 STRING,
>    PRIMARY KEY(id) NOT ENFORCED
> ) with (
>     'connector' = 'jdbc',
>     'url' = 'jdbc:mysql://172.16.11.43:4001/oggadmin',
>     'table-name' = 'users',
>     'driver' = 'com.mysql.cj.jdbc.Driver',
>     'username' = 'oggadmin',
>     'password' = 'oggadmin'
> );
8.将test1，test2表关联查询结果写入t4中，同时写回到tidb的表users中。
Flink SQL> insert into t4 select * from test1,test2 where test1.id=test2.id;
[INFO] Submitting SQL update statement to the cluster...
[INFO] SQL update statement has been successfully submitted to the cluster:
Job ID: 8c7565ce11890f8f7584bf6ff2d47b7e
9.查询users中表数据
mysql> select * from users;
+----+------+-----+-------+
| id | name | id0 | name0 |
+----+------+-----+-------+
|  1 | 1    |   1 | 1     |
|  2 | 2    |   2 | 2     |
|  3 | 3    |   3 | 3     |
|  4 | 4    |   4 | 4     |
|  5 | 5    |   5 | 5     |
|  6 | 6    |   6 | 6     |
|  7 | 7    |   7 | 7     |
|  8 | 8    |   8 | 8     |
|  9 | 9    |   9 | 9     |
| 10 | 10   |  10 | 10    |
| 11 | 11   |  11 | 11    |
| 12 | 12   |  12 | 12    |
| 13 | 13   |  13 | 13    |
| 14 | 14   |  14 | 14    |
+----+------+-----+-------+
14 rows in set (0.00 sec)
mysql> select * from users;
+----+------+-----+-------+
| id | name | id0 | name0 |
+----+------+-----+-------+
|  1 | 1    |   1 | 1     |
|  2 | 2    |   2 | 2     |
|  3 | 3    |   3 | 3     |
|  4 | 4    |   4 | 4     |
|  5 | 5    |   5 | 5     |
|  6 | 6    |   6 | 6     |
|  7 | 7    |   7 | 7     |
|  8 | 8    |   8 | 8     |
|  9 | 9    |   9 | 9     |
| 10 | 10   |  10 | 10    |
| 11 | 11   |  11 | 11    |
| 12 | 12   |  12 | 12    |
| 13 | 13   |  13 | 13    |
| 14 | 14   |  14 | 14    |
| 15 | 15   |  15 | 15    |
+----+------+-----+-------+
15 rows in set (0.01 sec)
```



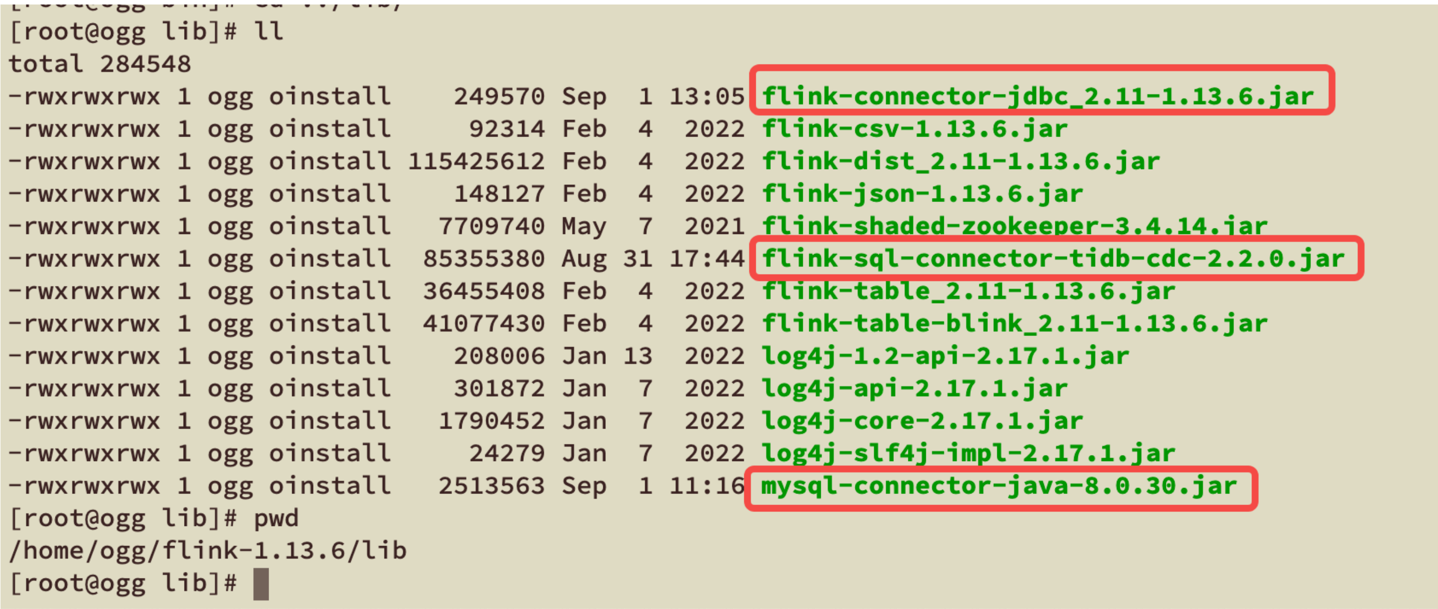
172.16.11.70:8081/#/job/8c7565ce11890f8f7584bf6ff2d47b7e/overview

Version: 1.13.6 | Commit: b2ca390 @ 2022-02-03T14:54:22+01:00 | Message:

insert-into_default_catalog.default_database.t4 RUNNING 5

ID: 8c7565ce11890f8f7584bf6ff2d47b7e | Start Time: 2022-09-01 14:07:51 | Duration: 17m 19s

不安全
新标签页
PingCAP
数据库
DP
大数据
Gmail
其他
Apache Flink Dashboard
Overview
Jobs
Running Jobs
Completed Jobs
Task Managers
Job Manager
Submit New Job
Cancel Job
Overview
Exceptions
TimeLine
Checkpoints
Configuration
Source: TableSourceScan(table=[[default_catalog, default_database, test1]], fields=[id, name])
Parallelism: 1
Backpressured (max): 0%
Busy (max): N/A
HASH
ChangelogNormalize(key=[id])
Parallelism: 1
Backpressured (max): 0%
Busy (max): 0%
HASH
Join(joinType=[InnerJoin], where=[(id = id0)], select=[id, name, id0, name0], leftInputSpec=[JoinKeyContainsUniqueKey], rightInputSpec=[JoinKeyContainsUniqueKey]) -> NotNullEnforcer(fields=[id]) -> Sink: Sink(table=[default_catalog.default_database.t4], fields=[id, name, id0, name0])
Parallelism: 1
Backpressured (max): 0%
Busy (max): 0%
Source: TableSourceScan(table=[[default_catalog, default_database, test2]], fields=[id, name])
Parallelism: 1
Backpressured (max): 0%
Busy (max): N/A
HASH
ChangelogNormalize(key=[id])
Parallelism: 1
Backpressured (max): 0%
Busy (max): 0%
HASH



```
[root@ogg bin]# cd ../lib/
[root@ogg lib]# ll
total 284548
-rwxrwxrwx 1 ogg oinstall    249570 Sep  1 13:05 flink-connector-jdbc_2.11-1.13.6.jar
-rwxrwxrwx 1 ogg oinstall     92314 Feb  4  2022 flink-csv-1.13.6.jar
-rwxrwxrwx 1 ogg oinstall 115425612 Feb  4  2022 flink-dist_2.11-1.13.6.jar
-rwxrwxrwx 1 ogg oinstall    148127 Feb  4  2022 flink-json-1.13.6.jar
-rwxrwxrwx 1 ogg oinstall   7709740 May  7  2021 flink-shaded-zookeeper-3.4.14.jar
-rwxrwxrwx 1 ogg oinstall  85355380 Aug 31 17:44 flink-sql-connector-tidb-cdc-2.2.0.jar
-rwxrwxrwx 1 ogg oinstall  36455408 Feb  4  2022 flink-table_2.11-1.13.6.jar
-rwxrwxrwx 1 ogg oinstall  41077430 Feb  4  2022 flink-table-blink_2.11-1.13.6.jar
-rwxrwxrwx 1 ogg oinstall    208006 Jan 13  2022 log4j-1.2-api-2.17.1.jar
-rwxrwxrwx 1 ogg oinstall    301872 Jan  7  2022 log4j-api-2.17.1.jar
-rwxrwxrwx 1 ogg oinstall   1790452 Jan  7  2022 log4j-core-2.17.1.jar
-rwxrwxrwx 1 ogg oinstall     24279 Jan  7  2022 log4j-slf4j-impl-2.17.1.jar
-rwxrwxrwx 1 ogg oinstall   2513563 Sep  1 11:16 mysql-connector-java-8.0.30.jar
[root@ogg lib]# pwd
/home/ogg/flink-1.13.6/lib
[root@ogg lib]# 
```