用户推荐数据从BigQuery导出到BigTable

BigQuery是谷歌云的数据仓库产品，主要用来执行数据分析任务，其延时一般在秒级。BigTable是谷歌云的NoSQL数据库产品，主要用来执行高并发实时查询和写入任务，其延时一般在10毫秒以下。结合二者优点，我们可以将针对用户的分析型处理在BigQuery中执行，结果存储在有用户ID字段的表中，然后用Dataflow导出数据到BigTable，存放在以用户ID作为Row Key的表中，之后就可以从客户端对BigTable中的数据做高速读取。下面的例子针对一个简单的内容推荐场景提供相应示例。

BigQuery数据表准备

可以准备一个符合以下Schema的表，存放示例数据。

```
[
  {
    "mode": "NULLABLE",
    "name": "predicted_rating",
    "type": "FLOAT"
  },
  {
    "mode": "NULLABLE",
    "name": "user_id",
    "type": "INTEGER"
  },
  {
    "mode": "NULLABLE",
    "name": "item_id",
    "type": "INTEGER"
  },
  {
    "mode": "NULLABLE",
    "name": "rating_rank",
    "type": "INTEGER"
  }
```

```
]
```

本例中的表为`youzhi-lab.movielens.recomm_ranked_filtered_short`，其权限设定为可以公共读取。其中内容如下。

Row	predicted_rating	user_id	item_id	rating_rank
1	4.8766766223891409	1	356	1
2	4.740690997059704	1	318	2
3	4.7170527143988785	1	1234	3
4	4.7052213124968878	1	2905	4
5	4.70433886514265	1	527	5
6	4.7937573047432629	2	356	1
7	4.6538003904722638	2	1240	2
8	4.644783312514762	2	2028	3
9	4.6304927464501473	2	110	4
10	4.5697576169622236	2	1246	5

其中各字段

- user_id：用户的ID
- item_id：推荐给用户的内容的ID
- predicted_rating：此条推荐内容的置信度
- rating_rank：此条推荐内容在给此用户的所有推荐内容中的置信度排名

BigTable数据表准备

在谷歌云控制台上创建好BigTable实例。本例的实例名为`bt-quickstart`，然后用谷歌云提供的cbt命令行工具创建表和列族。

```
cbt createtable user-recommendations
cbt createfamily user-recommendations cf_recommendations
```

表中不用写入任何数据。稍后作为目的表接收Dataflow写入的用户推荐数据。

运行Dataflow数据导入任务

首先下载代码到本地。

git clone https://github.com/cloudymoma/DataflowTemplates.git

创建一个谷歌云存储桶用作零时文件存储。本例为`youzhi-lab`。

注意：建议BigQuery数据集，BigTable实例，Dataflow临时文件存储桶和Dataflow集群都创建在同一个区域，这样会减少数据跨区域产生的延时和流量费用。

根据实际情况创建以下环境变量。其中高亮部分请用您项目实际内容填写。

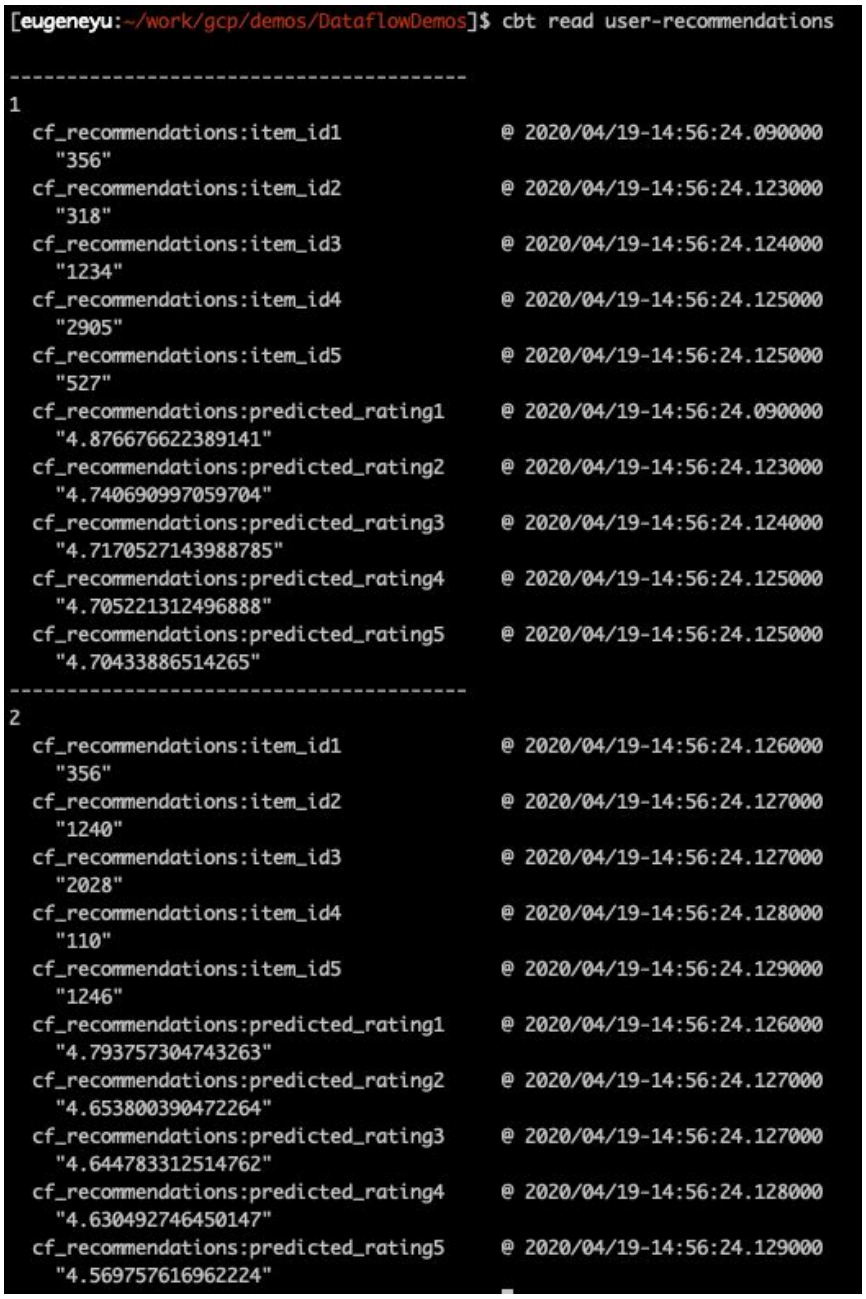
```
PROJECT_ID=youzhi-lab
BUCKET=youzhi-lab
BIGTABLE_INSTANCE_ID=bt-quickstart
BIGTABLE_TABLE=user-recommendations
BQ_TABLE=youzhi-lab:movielens.recomm_ranked_filtered_short
```

在本地编译运行。

```
mvn compile exec:java \
-Dexec.mainClass=com.google.cloud.teleport.templates.BigQueryToBigTable \
-Dexec.args=" \
--stagingLocation=gs://$BUCKET/dataflow/pipelines/staging \
--tempLocation=gs://$BUCKET/dataflow/pipelines/temp \
--bigtableProjectId=$PROJECT_ID \
--bigtableInstanceId=$BIGTABLE_INSTANCE_ID \
--bigtableTableId=$BIGTABLE_TABLE \
--bqTable=$BQ_TABLE"
```

确认输出没有错误，并且BigTable中接收到了数据。

```
cbt read user-recommendations
```

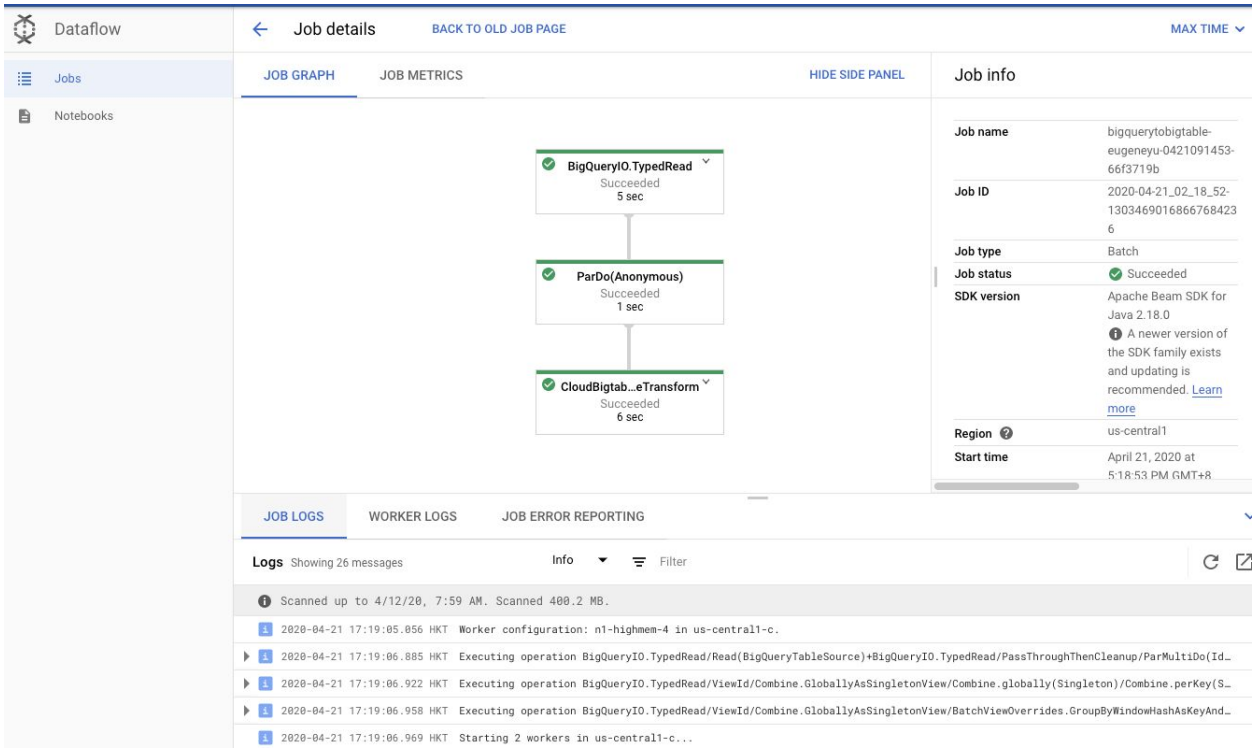


之后在提交Dataflow任务在谷歌云上运行。

```
mvn compile exec:java \
-Dexec.mainClass=com.google.cloud.teleport.templates.BigQueryToBigTable \
-Dexec.args=" \
--runner=DataflowRunner \
--workerMachineType=n1-highmem-4 \
--numWorkers=2 \
```

```
--region=us-central1 \
--stagingLocation=gs://$BUCKET/dataflow/pipelines/staging \
--tempLocation=gs://$BUCKET/dataflow/pipelines/temp \
--bigtableProjectId=$PROJECT_ID \
--bigtableInstanceId=$BIGTABLE_INSTANCE_ID \
--bigtableTableId=$BIGTABLE_TABLE \
--bqTable=$BQ_TABLE"
```

在谷歌云控制台查看Dataflow任务正确启动并且成功执行完成。确认没有任何错误。



之后在BigTable中查看数据已经更新。

到此，使用简单的Schema从BigQuery同步数据到BigTable成功完成。之后可以根据业务需求修改表的结构和数据，创建您自己的数据同步管道。如有必要，可以对Dataflow代码做简单修改之后编译运行。有更复杂的需求，也可以咨询谷歌云支持工程师。