目录

数据导入

Spark对接KS3

本文介绍Spark如何读取KS3中的数据。

背景信息

当前KMR：

- 支持通过免AccessKey方式访问KS3数据源。
- 支持通过显式写AccessKey和Endpoint方式访问KS3数据源。

Spark接入KS3示例

本示例为您展示，Spark如何以免AccessKey方式读取KS3中数据，并将处理完的数据写回至KS3。

```
val conf = new SparkConf().setAppName("Test ks3")
val sc = new SparkContext(conf)
val pathIn = "ks3://bucket/path/to/read"
val inputData = sc.textFile(pathIn)
val cnt = inputData.count
println(s"count: $cnt")
val outputPath = "ks3://bucket/path/to/write"
val outpuData = inputData.map(e => s"$e has been processed.")
outpuData.saveAsTextFile(outputPath)
```

PySpark接入KS3示例

本示例为您展示，PySpark如何以免AccessKey方式读取KS3中数据，并将处理完的数据写回至KS3。

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("Python Spark SQL ks3 example").getOrCreate()
pathIn = "ks3://bucket/path/to/read"
df = spark.read.text(pathIn)
cnt = df.count()
print(cnt)
outputPath = "ks3://bucket/path/to/write"
df.write.format("parquet").mode('overwrite').save(outputPath)
```

Hive开发手册

本文介绍如何在KMR集群中开发Hive作业流程。

在Hive中使用KS3

在Hive中读写KS3时，先创建一个external的表。

```
CREATE EXTERNAL TABLE eusers (
userid INT)
LOCATION 'ks3://emr/users';
```

当上面的方式无法支持，或者您希望使用非本账号的AccessKey来访问其他位置的KS3数据的时候，请使用如下方式。

```
CREATE EXTERNAL TABLE eusers (
userid INT)
LOCATION 'ks3://${AccessKeyId}:${AccessKeySecret}@${bucket}.${endpoint}/users';
```

参数说明：

- ${accessKeyId}：您账号的AccessKey ID。
- ${accessKeySecret}：该AccessKey ID对应的密钥。
- ${endpoint}：访问KS3使用的网络，由您集群所在的Region决定，对应的KS3也需要是在集群对应的Region。

使用示例

Hive作业流程示例如下：

- 示例1

1. 编写如下脚本，保存为hiveSample1.sql文件，并上传至KS3。

```
USE DEFAULT;
set hive.input.format=org.apache.hadoop.hive.ql.io.HiveInputFormat;
set hive.stats.autogather=false;
DROP TABLE emrusers;
CREATE EXTERNAL TABLE emrusers (
userid INT,
movieid INT,
rating INT,
unixtime STRING )
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
LOCATION 'ks3//${bucket}/yourpath';
SELECT COUNT(*) FROM emrusers;
SELECT * from emrusers limit 100;
SELECT movieid,count(userid) as usercount from emrusers group by movieid order by usercount desc limit 50;
```

2. 测试用数据资源 您可以下载如下Hive作业需要的资源，然后将其上传至您KS3对应的目录。 资源下载：公共测试数据。

3. 创建作业 在KMR中新建一个Hive作业，作业内容如下。

```
-f ks3ref://${bucket}/yourpath/hiveSample1.sql
```

其中${bucket}是您的KS3 Bucket，yourpath是Bucket上的路径，需要您填写实际保存Hive脚本的位置。

4. 运行作业 单击运行以运行作业。您可以关联一个已有的集群，也可以自动按需创建一个，然后关联上创建的作业。

- 示例2 以HiBench中的scan为例。

1. 编写如下脚本，上传至KS3。

```
USE DEFAULT;
set hive.input.format=org.apache.hadoop.hive.ql.io.HiveInputFormat;
set mapreduce.job.maps=12;
set mapreduce.job.reduces=6;
set hive.stats.autogather=false;
DROP TABLE uservisits;
CREATE EXTERNAL TABLE uservisits (sourceIP STRING,destURL STRING,visitDate STRING,adRevenue DOUBLE,userAgent STRING,country
Code STRING,languageCode STRING,searchWord STRING,duration INT ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS
 SEQUENCEFILE LOCATION 'ks3://${bucket}/sample-data/hive/Scan/Input/uservisits';
```

2. 准备测试数据 您可以通过下面的地址下载作业需要的资源，然后将其上传至您KS3对应的目录。

3. 在KMR中创建Hive作业，详情请参见Hive作业配置。
4. 运行作业 单击运行以运行作业。您可以关联一个已有的集群，也可以自动按需创建一个，然后关联上创建的作业。