使用Hive读取ElasticSearch中的数据

本文将介绍如何通过Hive来读取ElasticSearch中的数据，然后我们可以像操作其他正常Hive表一样，使用Hive来直接操作ElasticSearch中的数据，将极大的方便开发人员。本文使用的各组件版本分别为 Hive0.12、Hadoop-2.2.0、ElasticSearch 2.3.4。

我们先来看看ElasticSearch中相关表的mapping：

```
{
  "user": {
    "properties": {
      "regtime": {
        "index": "not_analyzed",
        "type": "string"
      },
      "uid": {
        "type": "integer"
      },
      "mobile": {
        "index": "not_analyzed",
        "type": "string"
      },
      "username": {
        "index": "not_analyzed",
        "type": "string"
      }
    }
  }
}
```

ElasticSearch中的index名为iteblog，type为user；user有regtime、uid、mobile以及username四个属性。现在我们在Hive端进行操作。

要让Hive能够操作ElasticSearch中的数据我们需要对Hive进行一些设置。值得高兴的是，ElasticSearch官方为我们提供了一些类库可以实现这些要求。我们需要引入相应的elasticsearch-hadoop-xxx.jar包，因为我们得ElasticSearch版本是2.x的，所以我们最少需要使用ES-Hadoop 2.2.x，本文使用的是elasticsearch-hadoop-2.3.4.jar，这个可以到Maven中央仓库下载。要让Hive能够加载elasticsearch-hadoop-2.3.4.jar文件有好几种方式：

1、直接通过add命令加载，如下：

```
hive > ADD JAR /home/iteblog/elasticsearch-hadoop-2.3.4.jar;
Added [/home/iteblog/elasticsearch-hadoop-2.3.4.jar] to class path
Added resources: [/home/iteblog/elasticsearch-hadoop-2.3.4.jar]
```

2、我们还可以在启动Hive的时候进行设置，如下：

```
$ bin/hive --auxpath=/home/iteblog/elasticsearch-hadoop-2.3.4.jar
```

3、我们还可以通过设置hive.aux.jars.path属性来实现：

```
$ bin/hive -hiveconf hive.aux.jars.path=/home/iteblog/elasticsearch-hadoop-2.3.4.jar
```

或者我们把这个设置直接写到hive-site.xml中，以便后面方便：

```
<property>
  <name>hive.aux.jars.path</name>
  <value>/home/iteblog/elasticsearch-hadoop-2.3.4.jar</value>
```

```
  <description>A comma separated list (with no spaces) of the jar files</description>
</property>
```

大家可以根据自己实际情况选择设置。设置好ElasticSearch相关类库之后，我们就可以到Hive中创建表了。为了方便，我们直接将Hive中各个字段以及类型设置成和ElasticSearch中一样：

```
hive (iteblog)> create EXTERNAL  table `user`(
              >   regtime string,
              >   uid int,
              >   mobile string,
              >   username string
              > )
              > STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
              > TBLPROPERTIES('es.resource' = 'iteblog/user', 'es.nodes'='www.iteblog.com', 'es.port'='9200', 'es.nodes.wan.only'='true');
```

到这里，我们已经已经可以在Hive里面查询ElasticSearch中的数据了：

```
hive (iteblog)> select * from  `user` limit 10;
OK
2016-10-24 13:08:16 1 13112121212 Tom
2016-10-24 14:08:16 2 13112121212 Join
2016-10-25 14:23:16 3 13112121212 iteblog
2016-10-25 13:08:16 4 NULL     weixin
2016-10-25 19:08:16 5 13112121212 bbs
2016-10-25 13:14:04 6 NULL     zhangshan
2016-10-25 13:08:16 7 13112121212 wangwu
2016-10-25 14:56:16 8 13112121212 Joan
2016-10-25 15:25:16 9 13112121212 White
2016-10-25 17:24:16 0 NULL     lihhh
Time taken: 0.072 seconds, Fetched: 10 row(s)
```

如上所述，我们已经成功通过Hive查询到ElasticSearch中的数据了。如果你在通过Hive查询ElasticSearch中的数据遇到如下异常：

```
Failed with exception java.io.IOException:org.elasticsearch.hadoop.EsHadoopIllegalArgumentException: Cannot detect ES version - typically this happens if the network/Elasticsearch cluster
```

is not accessible or when targeting a WAN/Cloud instance without the proper setting 'es.nodes.wan.only'

这个很可能是因为你配置错了 es.nodes 或者 es.port 属性了。

在上面的例子中，我们为了方便将Hive中的字段设置成和ElasticSearch中一样；但实际情况下，我们可能无法将Hive中的字段和ElasticSearch保持一致，这时候我们需要在创建Hive表的时候做一些设置，否则将会出现错误。我们可以通过 es.mapping.names 参数实现，如下：

```
hive (iteblog)> create EXTERNAL  table `user`(
              >  register_time string,
              >  user_id int,
              >  mobile string,
              >  username string
              > )
              > STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
              > TBLPROPERTIES('es.resource' = 'iteblog/user', 'es.nodes'='www.iteblog.com', 'es.port'='9200', 'es.nodes.wan.only'='true','es.mapping.names'='register_time:regtime,user_id:uid');
```

然后我们就可以将Hive中的 register_time 映射到ElasticSearch中的 regtime 字段； user_id 映射到ElasticSearch中的 uid 字段。

在创建Hive表的时候，我们还可以通过制定 es.query 来限制需要查询的数据，如下：

```
hive (iteblog)> create EXTERNAL  table `user`(
              >  regtime string,
              >  uid int,
              >  mobile string,
              >  username string
              > )
              > STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
              > TBLPROPERTIES('es.resource' = 'iteblog/user', 'es.nodes'='www.iteblog.com', 'es.port'='9200', 'es.nodes.wan.only'='true','es.query' = '?q=uid:2');
```

上面的查询仅返回uid为2的数据（关于查询条件设置可以参见《23种非常有用的ElasticSearch查询例子(1)》），然后我们可以看效果：

```
hive (iteblog)> select * from `user` limit 10;
OK
2016-10-24 14:08:16 2 13112121212 Join
Time taken: 0.023 seconds, Fetched: 1 row(s)
```

我们可以看到，uid为2的数据才返回了，其他的数据被过滤了。

在一些需要启动MapReduce任务来完成的SQL，Hive启动的Map个数和ElasticSearch中的分片个数一致，也就是每个分片使用一个Map任务来处理。