通过Hive将数据写入到ElasticSearch

我在《使用Hive读取ElasticSearch中的数据》文章中介绍了如何使用Hive读取ElasticSearch中的数据，本文将接着上文继续介绍如何使用Hive将数据写入到ElasticSearch中。在使用前同样需要加入 elasticsearch-hadoop-2.3.4.jar 依赖，具体请参见前文介绍。我们先在Hive里面建个名为iteblog的表，如下：

```
CREATE EXTERNAL TABLE iteblog (
   id      bigint,
   name    STRING)
STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
TBLPROPERTIES('es.resource' = 'iteblog/iteblog', 'es.nodes'='www.iteblog.com','es.port'='9003')
;
```

建完表之后我们可以看下Hive是怎么存储这样的表格：

```
hive> show create table iteblog;
OK
CREATE EXTERNAL TABLE `iteblog`(
  `id` bigint COMMENT 'from deserializer',
  `name` string COMMENT 'from deserializer')
```

```
ROW FORMAT SERDE
  'org.elasticsearch.hadoop.hive.EsSerDe'
STORED BY
  'org.elasticsearch.hadoop.hive.EsStorageHandler'
WITH SERDEPROPERTIES (
  'serialization.format'='1')
LOCATION
  'hdfs://user/iteblog/hive/warehouse/iteblog.db/iteblog'
TBLPROPERTIES (
  'COLUMN_STATS_ACCURATE'='false',
  'es.nodes'='www.iteblog.com',
  'es.port'='9003',
  'es.resource'='iteblog/iteblog',
  'numFiles'='0',
  'numRows'='-1',
  'rawDataSize'='-1',
  'totalSize'='0',
  'transient_lastDdlTime'='1478248148')
Time taken: 0.148 seconds, Fetched: 21 row(s)
```

我们可以看到Hive对里面的字段注释是 from deserializer，如果是正常的Hive表将没有这些信息；而且我们可以发现 ROW FORMAT SERDE 已经变成了 org.elasticsearch.hadoop.hive.EsSerDe ，在TBLPROPERTIES里面记录了一些链接ElasticSearch需要的参数配置。好了，现在我们在这个表里面导一些数据：

```
hive> insert into table iteblog select * from test limit 100;
```

上面的SQL运行完之后我们可以看到表所在的HDFS目录是没有数据的：

```
hive > dfs -ls /user/iteblog/hive/warehouse/iteblog.db/iteblog;
hive >
```

我们到ElasticSearch里面可以发现已经多了一个index和type，就是我们在建表时指定的es.resource，而且ElasticSearch为我们生成type的mapping如下：

```
{
   "iteblog": {
```

```
      "properties": {
        "name": {
          "type": "string"
        },
        "id": {
          "type": "long"
        }
      }
    }
}
```

这就Hive表里面的字段，类型都对应了。但是我们发现ElasticSearch中的 iteblog/iteblog 每行数据对应的id都是随机生成的，不过我们可以在建Hive表的时候加上 es.mapping.id 参数来指定我们自定义的id如下：

```
CREATE EXTERNAL TABLE iteblog (
   id      bigint,
   name    STRING)
STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
TBLPROPERTIES('es.resource' = 'iteblog/iteblog', 'es.nodes'='www.iteblog.com','es.port'='9003','
es.mapping.id' = 'id');
```

这样ElasticSearch中的 iteblog/iteblog 对应的id将会和Hive中的id字段一一对应。当然其他的字段也可以设置相应的mapping，可以通过 es.mapping.names 参数实现。

如何存Json数据

如果我们Hive里面的字段是Json数据，我们希望在ElasticSearch中解析这个json数据，然后在ElasticSearch中将解析的数据存起来，比如我们的Json数据格式为：{"id":"123","name":"iteblog"}，我们可以在建Hive表的时候加上 es.input.json 参数，这样ElasticSearch会解析这个json数据，如下：

```
CREATE EXTERNAL TABLE iteblog (
   json    STRING)
STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
TBLPROPERTIES('es.resource' = 'iteblog/iteblog', 'es.nodes'='www.iteblog.com','es.port'='9003','
es.input.json' = 'yes');
```

这样ElasticSearch为我们生成的mapping为：

```
{
  "iteblog": {
    "properties": {
      "name": {
        "type": "string"
      },
      "id": {
        "type": "string"
      }
    }
  }
}
```

而不是

```
{
  "iteblog": {
    "properties": {
      "json": {
        "type": "string"
      }
    }
  }
}
```

如果Hive中的数据是Json字段，但是在写ElasticSearch的时候使用了 es.input.json 配置，这时候在Hive里面查数会发现数据都是NULL：

```
hive > select * from iteblog limit 10;
OK
NULL
NULL
NULL
NULL
NULL
NULL
NULL
NULL
```

```
NULL
NULL
Time taken: 0.057 seconds, Fetched: 10 row(s)
```

数据为json的时候我们同样可以指定ElasticSearch的id生成的规则，如下：

```
CREATE EXTERNAL TABLE iteblog (
   json    STRING)
STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
TBLPROPERTIES('es.resource' = 'iteblog/iteblog', 'es.nodes'='www.iteblog.com','es.port'='9003','
es.input.json' = 'yes','es.mapping.id' = 'id');
```

这样就会把Json里面的id当作ElasticSearch中的id。

动态处理type

有时候我们可能希望根据数据的类别不一样来将数据存放到ElasticSearch中不同的type中，我们可以通过如下设置实现

```
CREATE EXTERNAL TABLE iteblog (
   id      bigint,
   name    STRING,
   type    STRING)
STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
TBLPROPERTIES('es.resource' = 'iteblog/{type}', 'es.nodes'='www.iteblog.com','es.port'='9003');
```

这样ElasticSearch会自动获取Hive中的type字段的值，然后将不同type的数据存放到ElasticSearch中不同的type中。如果Hive中的字段是json格式，比如 {"id":"123","name":"iteblog","type":"A"}，我们同样可以通过下面设置实现：

```
CREATE EXTERNAL TABLE iteblog (
   json      STRING)
STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'
TBLPROPERTIES('es.resource' = 'iteblog/{type}', 'es.nodes'='www.iteblog.com','es.port'='9003','
es.input.json' = 'yes');
```

这样ElasticSearch会自动为我们解析json中的type字段的值，然后决定将这条记录放到ElasticSearch中对应的type中。

Hive类型和ElasticSearch类型映射

Hive类型	Elasticsearch类型
void	null
boolean	boolean
tinyint	byte
smallint	short
int	int
bigint	long
double	double
float	float
string	string
binary	binary
timestamp	date
struct	map
map	map
array	array
union	目前不支持
decimal	string
date	date
varchar	string
char	string