

hugeGraph-Tools备份数据丢失实验报告V2

- 1、实验背景
- 2、实验基础配置环境
- 3、实验数据
- 4、实验过程
- 5、问题定位
- 6、特殊说明

1、实验背景

我们公司在业务环境中需要定期用hugeGraph-Tools对backend为scylla的hugegraph进行备份，每次备份完成都会发现有大概千分之2的数据丢失。

2、实验基础配置环境

jdk: 1.8

hugegraph: 0.11.2

backend: scylla 4.3.2 standalone

ps: 所有服务都是跑在windows10的笔记本电脑上，scylla运行在Oracle VM VirtualBox虚拟的centos7机器上，hugegraph运行在windows本地的idea中。

3、实验数据

- Schema & Load Test Data Java Code

```
1 public static void main(String[] args) throws Exception {
2     HugeClient hugeClient = HugeClient.builder("http://192.16
3     GraphManager graph = hugeClient.graph();
4     SchemaManager schema = hugeClient.schema();
5     schema.propertyKey("id").asText().ifNotExist().create();
6     schema.propertyKey("code").asText().ifNotExist().create()
7     ;
8     schema.vertexLabel("affiliation")
9     .properties("code").nullableKeys("code").useCusto
```

```

    mizeStringId().ifNotExist().create();
10
11     schema.vertexLabel("province")
12         .properties("code").nullableKeys("code").useCusto
    mizeStringId().ifNotExist().create();
13
14     schema.vertexLabel("author")
15         .properties("code").nullableKeys("code").useCusto
    mizeStringId().ifNotExist().create();
16
17     schema.vertexLabel("journal")
18         .properties("code").nullableKeys("code").useCusto
    mizeStringId().ifNotExist().create();
19
20     schema.vertexLabel("district")
21         .properties("code").nullableKeys("code").useCusto
    mizeStringId().ifNotExist().create();
22
23     schema.vertexLabel("corporationRaw")
24         .properties("code").nullableKeys("code").useCusto
    mizeStringId().ifNotExist().create();
25
26     schema.vertexLabel("city")
27         .properties("code").nullableKeys("code").useCusto
    mizeStringId().ifNotExist().create();
28
29     schema.vertexLabel("corporation")
30         .properties("code").nullableKeys("code").useCusto
    mizeStringId().ifNotExist().create();
31
32     schema.vertexLabel("paper")
33         .properties("code").nullableKeys("code").useCusto
    mizeStringId().ifNotExist().create();
34
35     schema.vertexLabel("cpc")
36         .properties("code").usePrimaryKeyId().primaryKeys
    ("code").ifNotExist().create();
37
38     schema.vertexLabel("patentTest")
39         .properties("code").nullableKeys("code").useCusto

```

```

    mizeStringId().ifNotExist().create();
40
41    schema.vertexLabel("patent")
42        .properties("code").nullableKeys("code").useCusto
    mizeStringId().ifNotExist().create();
43
44
45    schema.vertexLabel("ipc")
46        .properties("code").usePrimaryKeyId().primaryKeys
    ("code").ifNotExist().create();
47
48    schema.vertexLabel("sitc4")
49        .properties("code").usePrimaryKeyId().primaryKeys
    ("code").ifNotExist().create();
50
51    String filepath = "C:\\Users\\Hunter\\Desktop\\temp\\test
    Data";
52    BufferedReader reader = FileUtil.openBufferedReader(filep
    ath);
53    String temp;
54
55    while ((temp = reader.readLine()) != null){
56        JSONObject jsonLine = JSONObject.parseObject(temp);
57        JSONArray vertices = jsonLine.getJSONArray("vertices"
    );
58        for(Object o:vertices){
59            JSONObject jsonVertex = (JSONObject)o;
60
61            String label = jsonVertex.getString("label");
62            Object id = jsonVertex.get("id");
63            Vertex vertex = new Vertex(label);
64
65            if(label.equals("sitc4") || label.equals("ipc") || la
    bel.equals("cpc")){
66                JSONObject properties = jsonVertex.getJSONOb
    ject("properties");
67                vertex.property("code",properties.getString(
    "code"));
68            }else {
69                vertex.id(id);

```

```

70         }
71         graph.addVertex(vertex);
72     }
73 }
74 hugeClient.close();
75 FileUtil.closeReader(reader);
76 }

```

- 节点数据
testData.zip (json格式)

4、实验过程

- ①、首先启动hugeGraph-Tools的back功能，然后在log目录下获取到所有的shards信息。
- ②、遍历每一个shard信息，将shard中的start和end作为入参，分别对hugeGraph-Tools中com.baidu.hugegraph.manager.BackupManager类下的backupVertexShard方法和scylla#cqlsh进行调用，比较每一个shard在scylla端和hugeGraph端获取到的数据是否一致。
- ③、实验结果显示，当数据量超过10W的时候，hugeGraph会有丢失数据的现象，部分测试结果如下：

序号	start	end	scyllaCount	hugegraphCount
1	-8591328707958494 212	-85671839244213443 25	563863	561342
2	23780307226195184 97	240286591681485603 7	580828	577020
3	-1065521777928297 94	-74722224579077265	745275	740951
4	-1374928143467927 365	-13465170088280688 61	663736	660857
5	-2337581930956893 201	-23261536679904762 75	267567	266284

5、问题定位

通过对hugeGraph-Tools的
com.baidu.hugegraph.manager.BackupManager#backupVertexShard方法debug排查发现，

backupVertexShard在调用hugegraph中com.baidu.hugegraph.api.traversers.VerticesAPI#scan方法遇到shard中数据量比较大则采取分页策略，分页策略代码显示每批分页获取10W条的数据并且同时返回一个page分页信息。page分页信息的作用是在获取next分页数据时以不透明的方式传到hugegraph中进行获取。

在分页策略这里，hugegraph向scylla获取大量数据时是一个page一个page的获取，当hugegraph向scylla获取第一个10W条数据的时候，不断循环判断是否已经到达10W条，如果没有到达10W条则继续请求获取下一个page，如果到达10W条。则把10W条数据和当前page（这里我们称这个page为A page）的信息返回给hugegraph-Tools。此时就发生了丢数据的情况，因为最后一个page()的数据还没有完全给到hugegraph-Tools，换句话说hugegraph-Tools只请求了10W条，本A page中多余的数据就丢失了，下一次hugegraph-Tools携带上次请求得到的A page信息获取后续新的数据时，hugegraph会从A page的下一个page开始重新获取数据。这就造成了每一次分页都会在最后一个page丢失一部分数据。

6、特殊说明

①、测试数据来自生产数据库中的一个shard，start=-1976594182946969593，end=-1972231945949017720。

②、我们将hugegraph中com.baidu.hugegraph.backend.query.Query#reachLimit(long count)方法的

```
1 return count >= (limit + this.offset());
```

这行代码替换成

```
1 return false;
```

并把com.baidu.hugegraph.backend.store.cassandra.CassandraEntryIterator#fetch()方法的

```
1 while (this.remaining > 0 && this.rows.hasNext())
```

这行代码 替换成

```
1 int leftOfCurrentPage = this.results.getAvailableWithoutFetching()  
  ;  
2 while ((this.remaining > 0 || leftOfCurrentPage>0)&&this.rows.hasNext())
```

这两行，问题得以解决。

更改思路就是：在备份时不严格按照page_limit获取数据，而是把page_limit所在的最后一页数据都获取完。