

MapReduce 实验五报告

组号: 20

组长: 严德美 MF1733071

组员: 周天烁 MG1733099

李 论 MG1733027

裴俊宇 MG1633056

一、 实验内容

在给定的社交网络图中，统计图中所有三角形的数量。

一个社交网络可以看做是一张图。社交网络中的人对应于图的顶点；社交网络中的人际关系对应于图中的边。在本次实验任务中，只考虑一种关系，即用户之间的关注关系。

在统计前，需要先进行有向边到无向边的转换，依据如下逻辑转换：

IF ($A \rightarrow B$) OR ($B \rightarrow A$) THEN A-B

二、 设计思路

1. OR 逻辑：共有 3 个 MapReduce

第一趟 去重

两个邻接点 idA 和 idB 之间的关系为 link

Mapper1 <key,value>=<"idA-idB","link">

Reducer1 <key,value>=<"idA-idB","">

第二趟 建立搜索关系

Map 将单独一个顶点作为 key，value 为所有与这个 key 邻接的点；Reduce 遍历某个 key 的 value 中的所有点，设 key 和 value 中每个点的关系为 link，再寻找 value 中邻接的两个点，设为 search。

Mapper2 <key,value>=<"idA","idB">

Reducer2 <key,value>=<"idA-idB","link"> <key,value>=<"idA-idB","search">

第三趟 统计三角形个数

Mapper3 <key,value>=<"idA-idB","link"> <key,value>=<"idA-idB","search">

Reducer3 <key,value>=<"The sum of triangle in social networks is", result+" ">

2. AND 逻辑：共有 5 个 MapReduce

第一趟 去重

Mapper1 <key,value>=<"idA-idB", "">

Reducer1 <key,value>=<"idA-idB", "">

第二趟 建立搜索关系：寻找存在双向邻接关系的点 idA 和 idB

Mapper2 <key,value>=<"idA", "idB">

Reducer2 <key,value>=<"idA-idB", "link"> <key,value>=<"idB-idA", "search">

第三趟 变成无向图

Mapper3 <key,value>=<"idA-idB", "link"> <key,value>=<"idB-idA", "search">

Reducer3 <key,value>=<"idA", "idB">

第四趟 建立搜索关系

Mapper4 <key,value>=<"idA", "idB">

Reducer4 <key,value>=<"idA-idB", "link"> <key,value>=<"idA-idB", "search">

第五趟 统计三角形个数

Mapper5 <key,value>=<"idA-idB", "link"> <key,value>=<"idA-idB", "search">

Reducer5 <key,value>=<"The sum of triangle in social networks is", result+"">

三、 程序代码

1、 OR 逻辑

Mapper1 :

@Override

protected void map(LongWritable key, Text value, Context context) throws IOException,

InterruptedException {

String[] lines = value.toString().split("\\s+");

String idA = lines[0];

String idB = lines[1];

if(idA.compareTo(idB)>0){

String tmp = idA;

```

        idA = idB;
        idB = tmp;
        context.write(new Text(idA+"-"+idB),new Text("link"));
    }else if(idA.compareTo(idB)<0){
        context.write(new Text(idA+"-"+idB),new Text("link"));
    }
}

```

Reducer1:

```

private Text value = new Text();
@Override
protected void reduce(Text key, Iterable<Text> values, Context context) throws IOException,
    InterruptedException {
    value.set("");
    context.write(key,value);
}

```

Mapper2:

```

@Override
protected void map(LongWritable key, Text value, Context context) throws IOException,
    InterruptedException {
    Text idA = new Text() ;
    idA.set(value.toString().split("-")[0]);
    Text idB = new Text(value.toString().split("-")[1].trim());
    context.write(idA,idB);
}

```

Reducer2:

```

@Override
protected void reduce(Text key, Iterable<Text> values, Context context) throws
IOException, InterruptedException {
    ArrayList<String> endList = new ArrayList<String>();
    Text edge = new Text();
    Text rel = new Text();
    rel.set("link");
    for(Text value:values){
        endList.add(value.toString());
        edge.set(key.toString()+"-"+value.toString());
        context.write(edge,rel);
    }
}

```

```

rel.set("search");
for(int i = 0; i<endList.size();i++){
    for(int j = i+1; j<endList.size();j++){
        String a = endList.get(i);
        String b = endList.get(j);
        if(a.compareTo(b)<0){
            edge.set(a+"-"+b);

        }else{
            edge.set(b+"-"+a);
        }
        context.write(edge,rel);
    }
}
}

```

Mapper3:

```

@Override
protected void map(LongWritable key, Text value, Context context) throws IOException,
    InterruptedException {
    if(value.toString().trim().length()>0){
        String[] line = value.toString().split("\t");
        if(line.length >=2)
            context.write(new Text(line[0].trim()),new Text(line[1].trim()));
    }
}

```

Reducer3:

```

private static long result = 0;
@Override
protected void reduce(Text key, Iterable<Text> values, Context context) throws IOException,
    InterruptedException {
    long count = 0;
    boolean flag = false;
    for (Text value:values){
        if(value.toString().trim().equals("link")){
            flag = true;
        }
        else{
            count++;
        }
    }
    if(flag)result += count;
}

```

```
}
```

```
@Override  
protected void cleanup(Context context) throws IOException, InterruptedException {  
    context.write(new Text("The sum of triangle in social networks is"),new Text(result+""));  
}
```

2、 AND 逻辑

Mapper1:

```
@Override  
protected void map(LongWritable key, Text value, Context context) throws IOException,  
    InterruptedException {  
    String[] lines = value.toString().split("\\s+");  
    String idA = lines[0];  
    String idB = lines[1];  
    context.write(new Text(idA+"-"+idB),new Text(""));  
}
```

Reducer1:

```
private Text value = new Text();  
@Override  
protected void reduce(Text key, Iterable<Text> values, Context context) throws IOException,  
    InterruptedException {  
    value.set("");  
    context.write(key,value);  
}
```

Mapper2:

```
@Override  
protected void map(LongWritable key, Text value, Context context) throws IOException,  
    InterruptedException {  
    Text idA = new Text() ;  
    String[] tmp = value.toString().split("-");  
    idA.set(tmp[0].trim());  
    Text idB = new Text(tmp[1].trim());  
    context.write(idA,idB);  
}
```

Reducer2:

```
@Override  
protected void reduce(Text key, Iterable<Text> values, Context context) throws IOException,  
    InterruptedException {
```

```

        Text edge = new Text();
        Text rel = new Text();
        for(Text value:values){
            edge.set(key.toString()+"-"+value.toString());
            rel.set("link");
            context.write(edge,rel);
            edge.set(value.toString()+"-"+key.toString());
            rel.set("search");
            context.write(edge,rel);
            context.progress();
        }
    }
}

```

Mapper3:

```

@Override
protected void map(LongWritable key, Text value, Context context) throws IOException,
    InterruptedException {

    String[] line = value.toString().split("\t");
    if(line.length >=2)
        context.write(new Text(line[0].trim()),new Text(line[1].trim()));

}

```

Reducer3:

```

@Override
protected void reduce(Text key, Iterable<Text> values, Context context) throws IOException,
    InterruptedException {
    boolean isLink = false;
    boolean isSearch = false;
    for(Text value:values){
        String rel = value.toString();
        if(rel.equals("link")){
            isLink = true;
        }else if(rel.equals("search")){
            isSearch = true;
        }
    }
    if(isLink&&isSearch){
        String[] edge = key.toString().split("-");
        if(edge[0].compareTo(edge[1])<0){

```

```

        context.write(new Text(edge[0]),new Text(edge[1]));
    }
}
}

```

其他 map、reduce 代码和 OR 逻辑一样

四、运行结果

OR 结果统计：

数据集	三角形个数	集群运行时间
Twitter	13082506	21min37sec
Google+	没有运行成功	没有运行成功

AND 结果统计：

数据集	三角形个数	集群运行时间
Twitter	1818304	2min9sec
Google+	27018510	40min56sec

运行结果截图：

Twitter OR: resultOutput/part-r-00000

```

2017st20@master01:~/workspace
[2017st20@master01 workspace]$ hadoop fs -cat resultOutput/part-r-00000
17/11/18 10:12:25 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
The sum of triangle in social networks is      13082506
[2017st20@master01 workspace]$

```

Twitter AND: triangleCountAnd/part-r-00000

```

2017st20@master01:~/workspace
Bytes Written=50
[2017st20@master01 workspace]$ hadoop fs -ls
17/11/21 11:11:46 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin
-java classes where applicable
Found 8 items
drwxr-xr-x - 2017st20 hadoop_user      0 2017-11-20 23:31 gplus_Output
drwxr-xr-x - 2017st20 hadoop_user      0 2017-11-21 05:44 gplus_resultOutput
drwxr-xr-x - 2017st20 hadoop_user      0 2017-10-24 10:11 hdfsInput
drwxr-xr-x - 2017st20 hadoop_user      0 2017-10-28 17:26 hdfsOutput
drwxr-xr-x - 2017st20 hadoop_user      0 2017-11-17 22:29 resultOutput
drwxr-xr-x - 2017st20 hadoop_user      0 2017-11-21 11:08 triangleCountAnd
drwxr-xr-x - 2017st20 hadoop_user      0 2017-11-17 22:08 triangleOutput
drwxr-xr-x - 2017st20 hadoop_user      0 2017-11-21 11:02 triangleOutputAnd
[2017st20@master01 workspace]$ hadoop fs -cat triangleCountAnd/part-r-00000
17/11/21 11:12:59 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin
-java classes where applicable
The sum of triangle in social networks is      1818304
[2017st20@master01 workspace]$

```

Google+ OR: gplus_resultOutput/part-r-00000

第三趟没有运行成功，前两趟运行成功，前两趟结果分别在

gplus_Output/tmp/step1/part-r-00000

gplus_Output/tmp/step2/part-r-00000

```
2017st20@master01:~/workspace
100046418043149960706-108014509161911463231
100046418043149960706-108017463391867103666
100046418043149960706-108032623298499022371
100046418043149960706-108045106590963354026
100046418043149960706-108055899058145820867
100046418043149960706-108091356091811422205
100046418043149960706-108092487151722513439
100046418043149960706-108096298177137829845
100046418043149960706-108099300904077351421
100046418043149960706-108115600437259850260
100046418043149960706-108156225708934962939
100046418043149960706-108189481625587435470
100046418043149960706-108189587050871927619
100046418043149960706-108202298147606481274
100046418043149960706-108206369568878291159
100046418043149960706-108215644829092784557
100046418043149960706-108252669961575680987
100046418043149960706-108259617195878049851
100046418043149960706-108263581871732759312
100046418043149960706-108281923609340751312
100046418043149960706-108345315125027346749
100046418043149960706-108347134468694009774
100046418043149960706-108348497916675615120
100046418043149960706-108360574989017
```

```
2017st20@master01:~/workspace
108089733686836919426-108798853537183937826 search
108089733686836919426-108801899952694316017 search
108089733686836919426-108806077592664974320 search
108089733686836919426-108810107917992542483 search
108089733686836919426-108810157580311791513 search
108089733686836919426-108812325444589721927 search
108089733686836919426-108817515291499537774 search
108089733686836919426-108848018340439793410 search
108089733686836919426-108853564919963291707 search
108089733686836919426-108855511624675713653 search
108089733686836919426-108872036879266293791 search
108089733686836919426-108875451177923572450 search
108089733686836919426-108876877414135743596 search
108089733686836919426-108882384433541220209 search
108089733686836919426-108890457957443035199 search
108089733686836919426-108892475208618187879 search
108089733686836919426-108894439689698688014 search
108089733686836919426-108894901907715020153 search
108089733686836919426-108896060981319977655 search
108089733686836919426-108896108200209113002 search
108089733686836919426-108901599657368504920 search
108089733686836919426-108913146943579525206 search
108089733686836919426-108921168647330834431 search
108089733686836919426-1089236968
```

Google+ AND: gplus_CountAnd/part-r-00000


```
2017st20@master01:~/workspace
Virtual memory (bytes) snapshot=241845559296
Total committed heap usage (bytes)=85633531904
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=6472861195
File Output Format Counters
Bytes Written=51
[2017st20@master01 workspace]$ hadoop fs -cat gplus_CountAnd/part-r-00000
17/11/21 12:02:37 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin
-java classes where applicable
The sum of triangle in social networks is      27018510
[2017st20@master01 workspace]$
```

WebUI 执行报告:

Twitter OR:

application_1508726229114_1534	2017st20	graphTriangleCount.jar	MAPREDUCE	root.2017st20	Fri Nov 17 22:20:04 +0800 2017	Fri Nov 17 22:29:58 +0800 2017	FINISHED	SUCCEEDED	<div></div>	History
application_1508726229114_1531	2017st20	graphTriangleCount.jar	MAPREDUCE	root.2017st20	Fri Nov 17 22:10:01 +0800 2017	Fri Nov 17 22:19:52 +0800 2017	FINISHED	SUCCEEDED	<div></div>	History
application_1508726229114_1530	2017st20	graphTriangleCount.jar	MAPREDUCE	root.2017st20	Fri Nov 17 22:08:21 +0800 2017	Fri Nov 17 22:09:55 +0800 2017	FINISHED	SUCCEEDED	<div></div>	History






Twitter AND:

application_1508726229114_2075	2017st20	Job5	MAPREDUCE	root.2017st20	Tue Nov 21 11:06:25 +0800 2017	Tue Nov 21 11:08:26 +0800 2017	FINISHED	SUCCEEDED	<div></div>	History
application_1508726229114_2074	2017st20	Job4	MAPREDUCE	root.2017st20	Tue Nov 21 11:05:32 +0800 2017	Tue Nov 21 11:06:22 +0800 2017	FINISHED	SUCCEEDED	<div></div>	History
application_1508726229114_2073	2017st20	Job3	MAPREDUCE	root.2017st20	Tue Nov 21 11:03:51 +0800 2017	Tue Nov 21 11:05:24 +0800 2017	FINISHED	SUCCEEDED	<div></div>	History
application_1508726229114_2072	2017st20	Job2	MAPREDUCE	root.2017st20	Tue Nov 21 11:03:04 +0800 2017	Tue Nov 21 11:03:48 +0800 2017	FINISHED	SUCCEEDED	<div></div>	History
application_1508726229114_2071	2017st20	Job1	MAPREDUCE	root.2017st20	Tue Nov 21 11:02:17 +0800 2017	Tue Nov 21 11:03:01 +0800 2017	FINISHED	SUCCEEDED	<div></div>	History

Google+ OR:

application_1508726229114_2048	2017st20	graphTriangleCount.jar	MAPREDUCE	root.2017st20	Tue Nov 21 05:44:06 +0800 2017	N/A	RUNNING	UNDEFINED	<div></div>	ApplicationMaster
application_1508726229114_2034	2017st20	graphTriangleCount.jar	MAPREDUCE	root.2017st20	Mon Nov 20 23:34:43 +0800 2017	Tue Nov 21 05:44:03 +0800 2017	FINISHED	SUCCEEDED	<div></div>	History
application_1508726229114_2033	2017st20	graphTriangleCount.jar	MAPREDUCE	root.2017st20	Mon Nov 20 23:30:47 +0800 2017	Mon Nov 20 23:34:30 +0800 2017	FINISHED	SUCCEEDED	<div></div>	History

Google+ AND

application_1508726229114_2082	2017st20	Job5	MAPREDUCE	root.2017st20	Tue Nov 21 11:47:05 +0800 2017	Tue Nov 21 12:01:45 +0800 2017	FINISHED	SUCCEEDED		History
application_1508726229114_2079	2017st20	Job4	MAPREDUCE	root.2017st20	Tue Nov 21 11:38:10 +0800 2017	Tue Nov 21 11:47:00 +0800 2017	FINISHED	SUCCEEDED		History
application_1508726229114_2078	2017st20	Job3	MAPREDUCE	root.2017st20	Tue Nov 21 11:30:36 +0800 2017	Tue Nov 21 11:38:06 +0800 2017	FINISHED	SUCCEEDED		History
application_1508726229114_2077	2017st20	Job2	MAPREDUCE	root.2017st20	Tue Nov 21 11:25:21 +0800 2017	Tue Nov 21 11:30:29 +0800 2017	FINISHED	SUCCEEDED		History
application_1508726229114_2076	2017st20	Job1	MAPREDUCE	root.2017st20	Tue Nov 21 11:20:49 +0800 2017	Tue Nov 21 11:25:18 +0800 2017	FINISHED	SUCCEEDED		History

五、性能分析、不足和改进

由于此次实验数据量较大，运行过程比较慢，中间结果的数据量应该尽量简化。此次实验运行速度较慢，还会出现运行到某一趟 MapReduce 时失败的情况，中途失败可以不用重新运行，直接从失败的那一趟 MapReduce 开始运行，读取之前已经写好中间结果。

六、实验感想

这次实验是实现社交网络图的三角形计数，在实验过程中，学会了使用多趟 MapReduce 程序串行进行处理数据，逐渐熟悉了 MapReduce 处理数据的框架和过程，对以后的使用大有益处。