

实验四 SecendSort

171180554 夏宇 171860617 肖维城

1. 实验设计

实验任务是使用MapReduce完成对数据的二次排序，输入数据为两列，先对第一列数字按照升序排列，即所谓的分组，再在每组中按照第二列数组进行降序排序完成二次排序。

数据结构：

使用自定义数据类型Data对数据进行包装，其构成为：

```
private int a;//第一个元素
private int b;//第二个元素
```

整体流程思路：

Map阶段：

输入输出：

输入<Key, Value>为：LongWritable, Text;

输出<Key, Value>为：Data, NullWritable（用作占位符）。

具体过程：

1. 使用job.**setInputFormatClass**(TextInputFormat.class)做为输入格式
2. 进入Mapper的map()方法，生成一个List，输入文件中的数据包装为Data，输出的**Key**和**Value**类型分别为：Data，NullWritable。
3. 在map阶段的最后，会先调用job.**setPartitionerClass**(DataPartitioner.class)对Map的结果进行分区，目的是将第一个值相同的数据分到同一个reducer上。
4. 每个分区内中又调用job.**setSortComparatorClass**()设置的key比较函数类排序，实验中没有设置，所以使用Data的实现的**compareTo**方法对输出的结果做二次排序，对数据中的第一个值进行升序排序，对第一个值相同的，按照第二个值进行降序排序。

Reduce阶段：

输入输出：

输入<Key, Value>为：Data, NullWritable;

输出<Key, Value>为：Data, NullWritable。

具体过程：

1. shuffle阶段：reducer开始获取所有映射到这个reducer的map输出值。
2. 构造一个key对应的value迭代器，这里需要使用job.**setGroupingComparatorClass**()设置的分组函数类。分组中，只需要Data里面的第一个值a相同，便将这些key分为同一组，将它们放在一个迭代器中。
3. 最后进入Reducer的**reduce()**方法，在迭代器中，依次进行write操作，得到输出。

2. 实验代码

Data

Data实现WritableComparator接口：定义compareTo函数：

```
public int compareTo(Data data) {
    //对数据中的第一个值进行升序排序，对第一个值相同的，按照第二个值进行降序排序
    if(a == data.a){
        return (-Integer.compare(b,data.b));
    }else{
        return Integer.compare(a,data.a);
    }
}
```

Mapper

```
public class SortMapper extends Mapper<LongWritable, Text, Data, NullWritable> {
    private Data data;

    @Override
    protected void setup(Context context)
        throws IOException, InterruptedException {
        //super.setup(context);
        //实例化Data
        data = new Data();
    }

    @Override
    protected void map(LongWritable key, Text value, Context context)
        throws IOException, InterruptedException {
        //super.map(key, value, context);
        //按照文件对data进行赋值
        String[] fields = value.toString().split("\t");
        data.setA(Integer.parseInt(fields[0]));
        data.setB(Integer.parseInt(fields[1]));
        context.write(data,NullWritable.get());
    }
}
```

partitioner

```
public class SortPartitioner extends HashPartitioner<Data, NullWritable> {
    @Override
    public int getPartition(Data key, NullWritable value, int numReduceTasks) {
        //按照第一个值a进行分组，将a相同的分到同一个reduce处理
        return (key.getA() & 2147483647) % numReduceTasks;
    }
}
```

Grouping

```
public int compare(WritableComparable a, WritableComparable b) {  
    //当a相同时，就认为是一组  
    Data da = (Data) a;  
    Data db = (Data) b;  
    return Integer.compare(da.getA(), db.getA());  
}
```

Reducer

```
public class SortReducer extends Reducer<Data, NullWritable, Data, NullWritable>  
{  
    @Override  
    protected void reduce(Data key, Iterable<NullWritable> values, Context  
context)  
        throws IOException, InterruptedException {  
        //利用迭代器读取、出路各个group下的的key  
        Iterator<NullWritable> iterator = values.iterator();  
        iterator.next();  
        while (iterator.hasNext()) {  
            context.write(key, NullWritable.get());  
            iterator.next();  
        }  
    }  
}
```

reference

参考网站：

<https://blog.csdn.net/lzm1340458776/article/details/42875751>
<https://zhuanlan.zhihu.com/p/65895097>

3. 实验结果

在编译完成jar包 `SecendSort.jar` 后，使用命令进行二次排序任务。

实验结果存放在 `/user/2020st05/exp4/output` 中。

部分内容如下所示：


```
[2020st05@master001 ~]$ hadoop fs -cat /user/2020st05/exp4/output/*
1      99
1      99
1      99
1      96
1      96
1      96
1      95
1      95
1      95
1      94
1      94
1      94
1      94
1      94
1      93
1      93
1      93
1      84
1      84
1      84
1      83
1      83
1      83
```

4. WebUI 执行报告

在大数据实验平台的历史记录中，查找SecendSort任务运行的记录

application_1572597966684_3934	2020st05	SecondSort	MAPREDUCE	root.team05	Sat May 9 11:35:14 +0800 2020	Sat May 9 11:35:31 +0800 2020	FINISHED	SUCCEEDED	History	N/A
--------------------------------	----------	------------	-----------	-------------	-------------------------------	-------------------------------	----------	-----------	-------------------------	-----

详细结果为：



MapReduce Job job_1572597966684_3934

Logged in as: dr.winc

- Application
- Job
 - Overview
 - Counters
 - Configuration
 - Map Tasks
 - Reduce Tasks
- Tools

Job Overview

Job Name: SecondSort
User Name: 2020st05
Queue: root.team05
State: SUCCEEDED
Uberized: false
Submitted: Sat May 09 11:35:14 CST 2020
Started: Sat May 09 11:35:37 CST 2020
Finished: Sat May 09 11:35:49 CST 2020
Elapsed: 11sec

Diagnostics:
Average Map Time: 2sec
Average Shuffle Time: 0sec
Average Merge Time: 0sec
Average Reduce Time: 3sec

ApplicationMaster		Start Time	Node	Logs
Attempt Number	2	Sat May 09 11:35:34 CST 2020	slave008:8042	logs

Task Type	Total	Complete
Map	1	1
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	0	1
Reduces	0	0	1

5. 实验分工

夏宇同学负责代码编写，编译jar包并在本机测试，完成实验报告。

肖维城同学负责代码编写，编译jar包并在本机、平台测试，在平台上得到结果。