大数据开发技术

东北林业大学

卢洋

第一章

MapReduce概述

1.7

MapReduce编程规范

用户编写的程序分为三个部分:

1. Mapper;
2. Reducer;
3. Driver.

- 固定格式，固定规范.

1 Mapper阶段

(1) 用户定义的Mapper要继承自己的父类;

(2) Mapper的输入数据是KV对的形式(KV的类型可自定义);

(3) Mapper中的业务逻辑写在map()方法中;

每一行一次map

(4) Mapper的输出数据是KV对的形式(KV的类型可自定义);

(5) map()方法(MapTask进程)对每一个<K, V>调用一次.

1 Mapper阶段

```
public static class TokenizerMapper
     extends Mapper<Object, Text, Text, IntWritable>{

  private final static IntWritable one = new IntWritable(1);
  private Text word = new Text();

  public void map(Object key, Text value, Context context
                  ) throws IOException, InterruptedException {
    StringTokenizer itr = new StringTokenizer(value.toString());
    while (itr.hasMoreTokens()) {
      word.set(itr.nextToken());
      context.write(word, one);
    }
  }
}
```

2 Reducer阶段

(1) 用户定义的Reducer要继承自己的父类;

(2) Reducer的输入数据类型对应Mapper的输出数据类型，也是KV对的形式;

(3) Reducer的业务逻辑写在reduce()方法中;

(4) ReduceTask进程对每一组相同K的<K, V>对调用一次reduce()方法.

2 Reducer阶段

```
public static class IntSumReducer
     extends Reducer<Text,IntWritable,Text,IntWritable> {
  private IntWritable result = new IntWritable();

  public void reduce(Text key, Iterable<IntWritable> values,
                     Context context
                     ) throws IOException, InterruptedException {
    int sum = 0;
    for (IntWritable val : values) {
      sum += val.get();
    }
    result.set(sum);
    context.write(key, result);
  }
```

3 Driver阶段

不用记住什么部分

- 相当于YARN集群的客户端；
- 用于提交整个程序到YARN集群；
- 提交的是封装了MapReduce程序相关运行参数的Job对象。