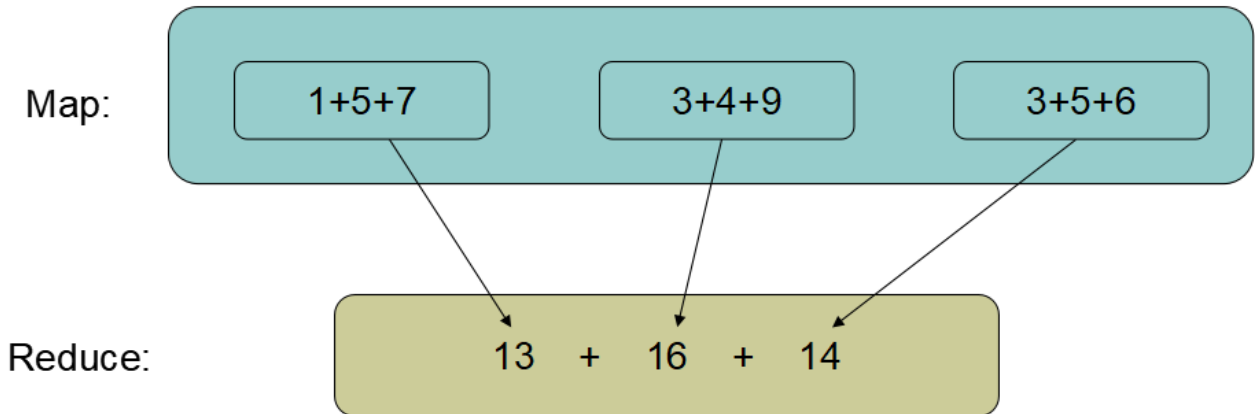


Hadoop MapReduce编程模型

问题提出

怎样解决海量数据的计算？

求和：1+5+7 +3+4+9 +3+5+6=?



MapReduce概述

Hadoop Map/Reduce是一个使用简易的软件框架，基于它写出来的应用程序能够运行在由上千个商用机器组成的大型集群上，并以一种可靠容错的方式并行处理上T级别的数据集。

一个Map/Reduce 作业（job）通常会把输入的数据集切分为若干独立的数据块，由 map任务（task）以完全并行的方式处理它们。框架会对map的输出先进行排序，然后把结果输入给reduce任务。通常作业的输入和输出都会被存储在文件系统中。整个框架负责任务的调度和监控，以及重新执行已经失败的任务。

通常，Map/Reduce框架和分布式文件系统是运行在一组相同的节点上的，也就是说，计算节点和存储节点通常在一起。这种配置允许框架在那些已经存好数据的节点上高效地调度任务，这可以使整个集群的网络带宽被非常高效地利用。

Map/Reduce框架由一个单独的master JobTracker 和每个集群节点一个slave TaskTracker共同组成。master负责调度构成一个作业的所有任务，这些任务分布在不同的slave上，master监控它们的执行，重新执行已经失败的任务。而slave仅负责执行由master指派的任务。

应用程序至少应指明输入/输出的位置（路径），并通过实现合适的接口或抽象类提供map和reduce函数。再加上其他作业的参数，就构成了作业配置（job configuration）。然后，Hadoop的job client提交作业（jar包/可执行程序等）和配置信息给JobTracker，后者负责分发这些软件和配置信息给slave、调度任务并监控它们的执行，同时提供状态和诊断信息给job-client。

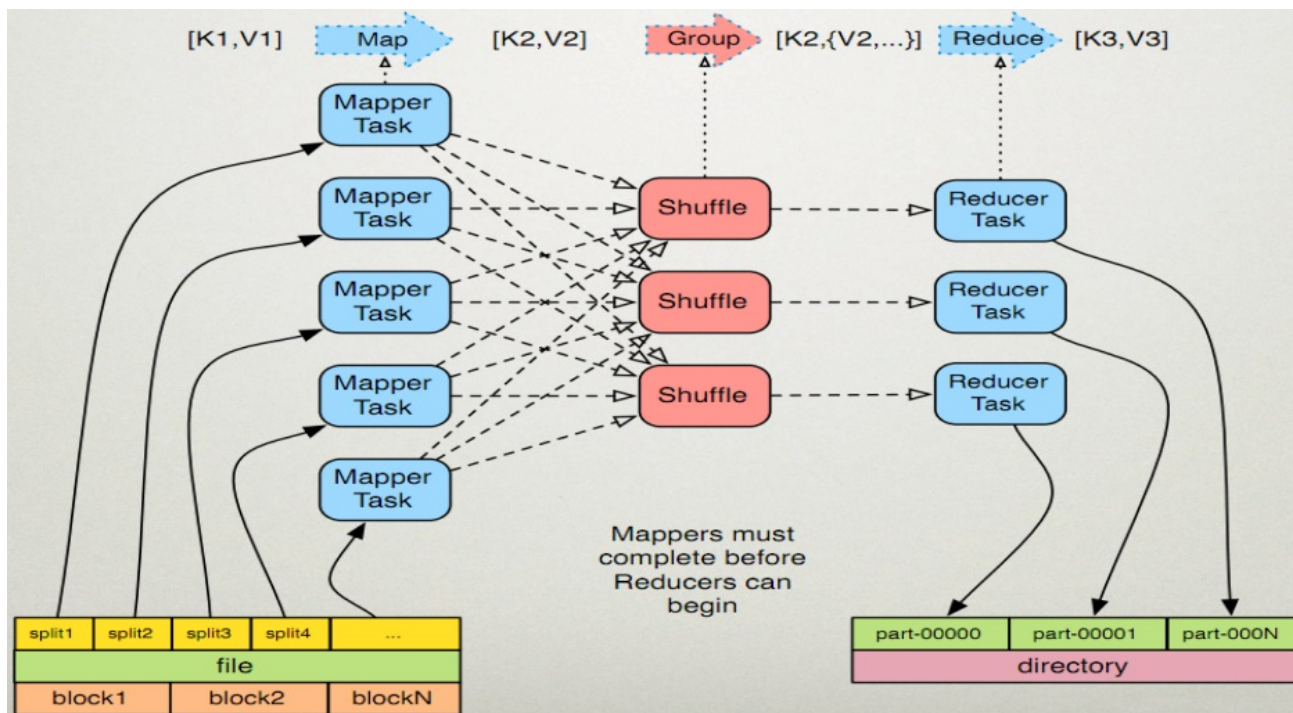
Mapreduce原理

Map/Reduce框架运转在<key, value> 键值对上，也就是说，框架把作业的输入看为是一组<key, value> 键值对，同样也产出一组<key, value> 键值对做为作业的输出，这两组键值对的类型可能不同。

框架需要对key和value的类(classes)进行序列化操作，因此，这些类需要实现 Writable接口。另外，为了方便框架执行排序操作，key类必须实现 WritableComparable接口。

一个Map/Reduce 作业的输入和输出类型如下所示：

(input) <k1, v1> -> map -> <k2, v2> -> combine -> <k2, v2> -> reduce -> <k3, v3> (output)



执行步骤：

1. map任务处理

- 1.1 读取输入文件内容，解析成key、value对。对输入文件的每一行，解析成key、value对。每一个键值对调用一次map函数。
- 1.2 写自己的逻辑，对输入的key、value处理，转换成新的key、value输出。
- 1.3 对输出的key、value进行分区。
- 1.4 对不同分区的数据，按照key进行排序、分组。相同key的value放到一个集合中。
- 1.5 (可选)分组后的数据进行归约。

2.reduce任务处理

- 2.1 对多个map任务的输出，按照不同的分区，通过网络copy到不同的reduce节点。
- 2.2 对多个map任务的输出进行合并、排序。写reduce函数自己的逻辑，对输入的key、value处理，转换成新的key、value输出。
- 2.3 把reduce的输出保存到文件中。

map、reduce键值对格式：

| 函数 | 输入键值对 | 输出键值对 |
|----------|-----------|---------|
| map() | <k1,v1> | <k2,v2> |
| reduce() | <k2,{v2}> | <k3,v3> |

MR Word Count程序

定义Mapper，需要继承Mapper类

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

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

    @Override
    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);
        }
    }
}

```

定义Reducer , 继承Reducer类

```

public class WordCountReducer extends Reducer<Text,IntWritable,Text,IntWritable> {

    private IntWritable result = new IntWritable();

    @Override
    protected 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);
    }
}

```

定义Job

```

Configuration conf = new Configuration();
Job job = Job.getInstance(conf, "word count");
job.setJarByClass(WordCountJob.class);
job.setMapperClass(WordCountMapper.class);
job.setCombinerClass(IntSumReducer.class);
job.setReducerClass(IntSumReducer.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
System.exit(job.waitForCompletion(true) ? 0 : 1);

```

提交作业

```

cd /home/opt/hadoop/hadoop-2.9.0/
bin/hadoop jar test/study-hadoop.jar com.myj.study.mr.WordCountJob /user/test /user/test2/wordcount

```

运行结果

Browse Directory

Go!

Show25▼entries

Search:

| <input type="checkbox"/> | Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name | <input type="checkbox"/> |
|--------------------------|------------|-------|------------|---------|---------------|-------------|------------|--------------|--------------------------|
| <input type="checkbox"/> | -rw-r--r-- | root | supergroup | 0 B | Jan 22 17:20 | 1 | 128 MB | _SUCCESS | <input type="checkbox"/> |
| <input type="checkbox"/> | -rw-r--r-- | root | supergroup | 28.1 KB | Jan 22 17:20 | 1 | 128 MB | part-r-00000 | <input type="checkbox"/> |

Showing 1 to 2 of 2 entries

Previous

1

Next

hadoop fs -tail /user/test2/wordcount/part-r-00000

```
[root@myj01 hadoop-2.9.0]# hadoop fs -tail /user/test2/wordcount/part-r-00000
ons 1
replication 1
replicationRecheckInterval 1
report 1
request 1
required 1
reside 1
restrictions 1
retry 1
root 1
safe 1
save 1
scan 1
scheduler: 1
second(s). 1
seconds. 1
secrets. 1
secs 1
security 1
see 1
segment 1
segments 1
selected. 1
server 1
services 1
set 2
signal 1
snapshot 1
software 2
```