



深入Hadoop 2.x（二）

谭唐华

课程大纲

- MapReduce Java API操作流程
- MapReduce 之 Wordcount实例
- MapReduce Shuffle
- MapReduce 调优

MapReduce 八股文

- ◆ MapReduce中，map和reduce函数遵循如下常规格式：

map: $(K_1, V_1) \rightarrow \text{list}(K_2, V_2)$

reduce: $(K_2, \text{list}(V_2)) \rightarrow \text{list}(K_3, V_3)$

- ◆ Mapper的基类：

```
protected void map(KEY key, VALUE value,  
    Context context) throws IOException, InterruptedException {  
}
```

- ◆ Reducer的基类：

```
protected void reduce(KEY key, Iterable<VALUE> values,  
    Context context) throws IOException, InterruptedException {  
}
```

- ◆ Context是上下文对象

MapReduce 八股文

```
Class MR{
Mapper☒  static public Class Mapper ...{
           //Map代码块
        }
Reducer☒ static public Class Reducer ...{
           //Reduce代码块
        }
Driver☒  main(){
           Configuration conf = new Configuration();
           Job job = new Job(conf, "job name");
           job.setJarByClass(this.MainClass.class);
           job.setMapperClass(Mapper.class);
           job.setReducerClass(Reducer.class);
           FileInputFormat.addInputPaths(job, new Path(args[0]));
           FileOutputFormat.setOutputPath(job, new Path(args[1]));
           //其他配置参数代码
           job.waitForCompletion(true);
        }
    }
```

◆ MapReduce将整个并行计算过程抽象到两个函数

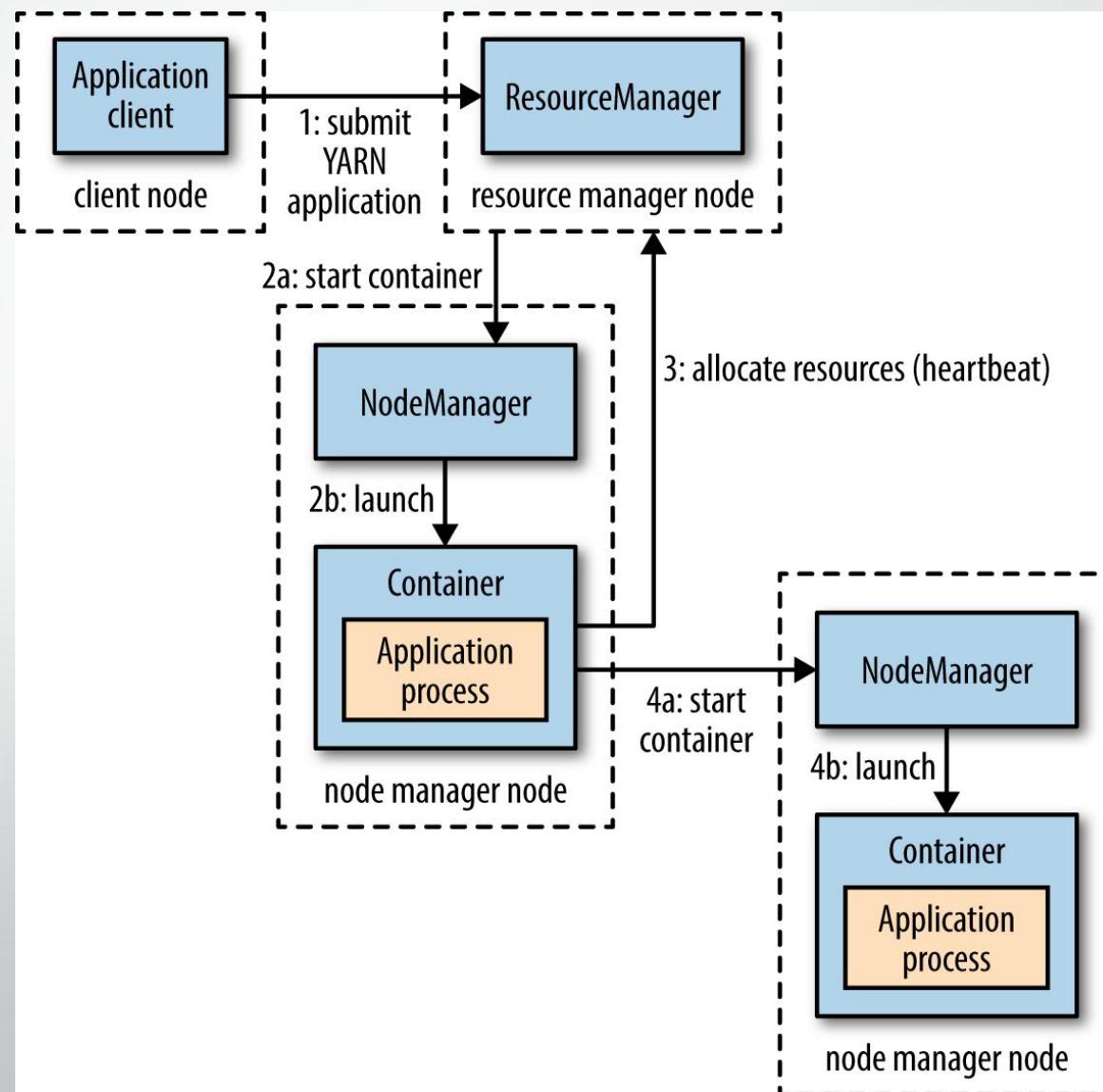
➤ Map(映射)：对一些独立元素组成的列表的每一个元素进行指定的操作，可以高度并行。

➤ Reduce(化简)：对一个列表的元素进行合并。

◆ 一个简单的MapReduce程序只需要指定map()、reduce()、input和output，剩下的事由框架完成。

MAPREDUCE ON YARN

- 1、客户端向ResourceManager提交任务。
- 2、ResourceManager向NodeManager分配一个ApplicationsManager，并找到一个容器生成一个MR App Mstr。
- 3、该应用的管理者向ResourceManager申请资源。
- 4、资源申请完毕后，找到NodeManager将容器中的MR App Mstr启动起来。
- 5、Map任务和Reduce任务启动。
- 6、Map与Reduce程序运行过程中都要向MR App Mstr提交信息。
- 7、当程序运行结束时，应用管理者向ResourceManager提交信息。

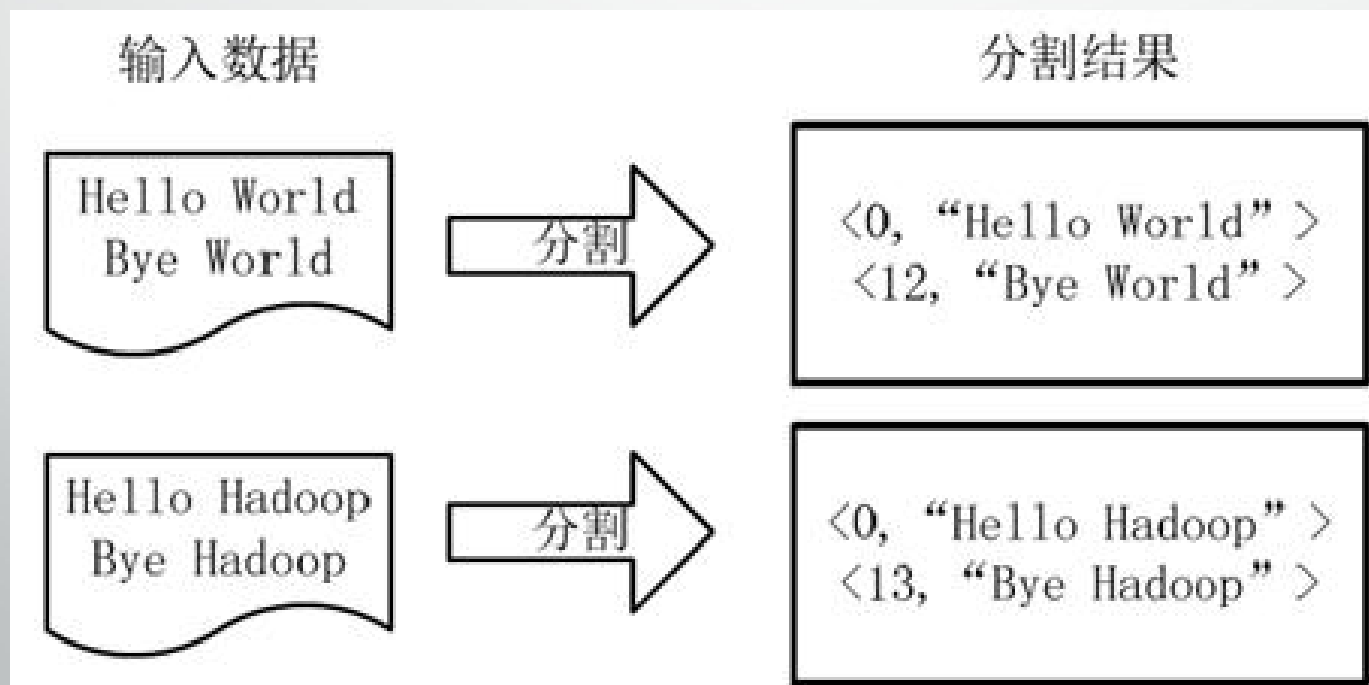


课程大纲

- MapReduce Java API操作流程
- MapReduce 之 Wordcount实例
- MapReduce 调优

WordCount 处理过程

- 将文件拆分成splits，由于测试用的文件较小，所以每个文件为一个split，并将文件按行分割形成<key,value>对，下图所示。这一步由MapReduce框架自动完成，其中偏移量（即key值）包括了回车所占的字符数（Windows/Linux环境不同）。



WordCount 处理过程

- 将分割好的<key,value>对交给用户定义的map方法进行处理，生成新的<key,value>对，下图所示。

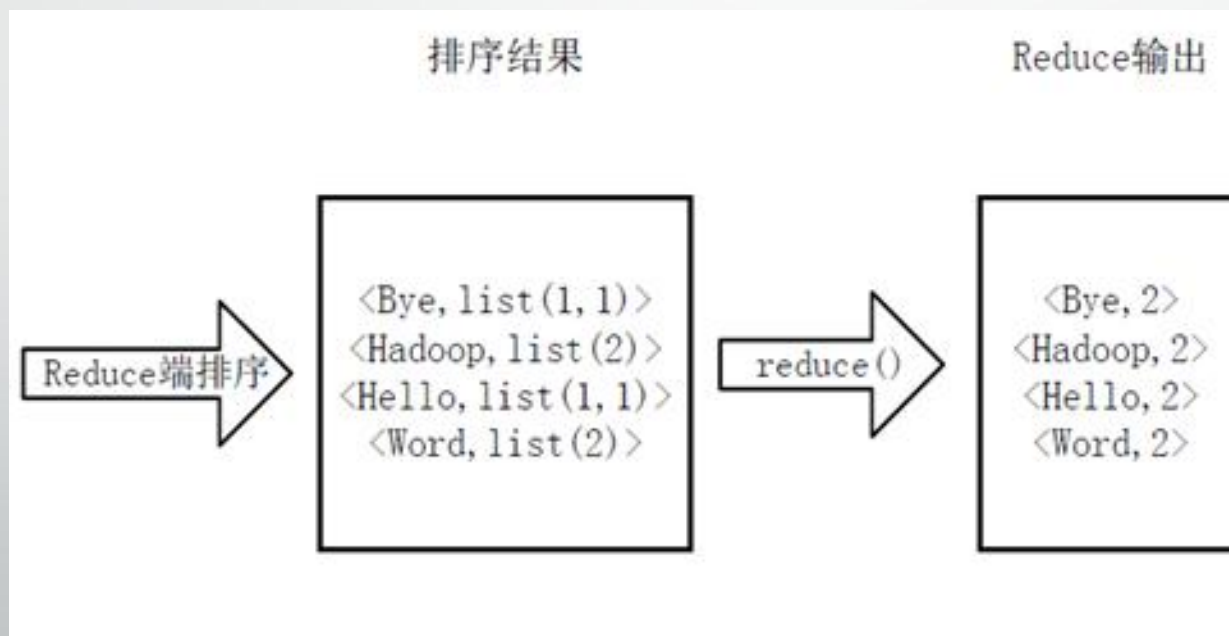


WordCount 处理过程

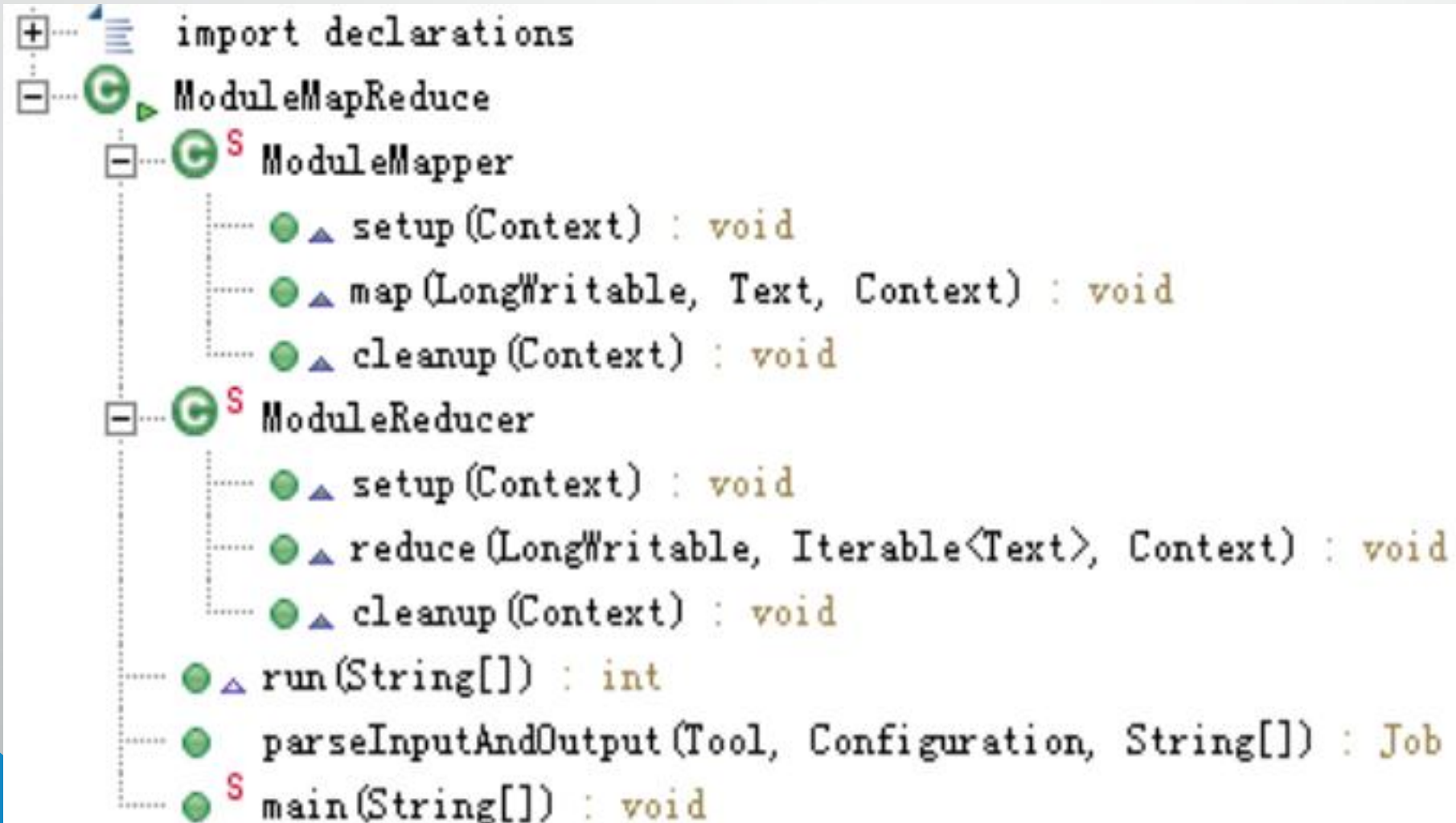
- 得到map方法输出的<key,value>对后，Mapper会将它们按照key值进行排序，得到Mapper的最终输出结果。



- Reducer先对从Mapper接收的数据进行排序、分组，再交由用户自定义的reduce方法进行处理，得到新的<key,value>对，并作为WordCount的输出结果，



MapReduce 编程模板类结构图



HADOOP: MapReduce 单词统计案例编程

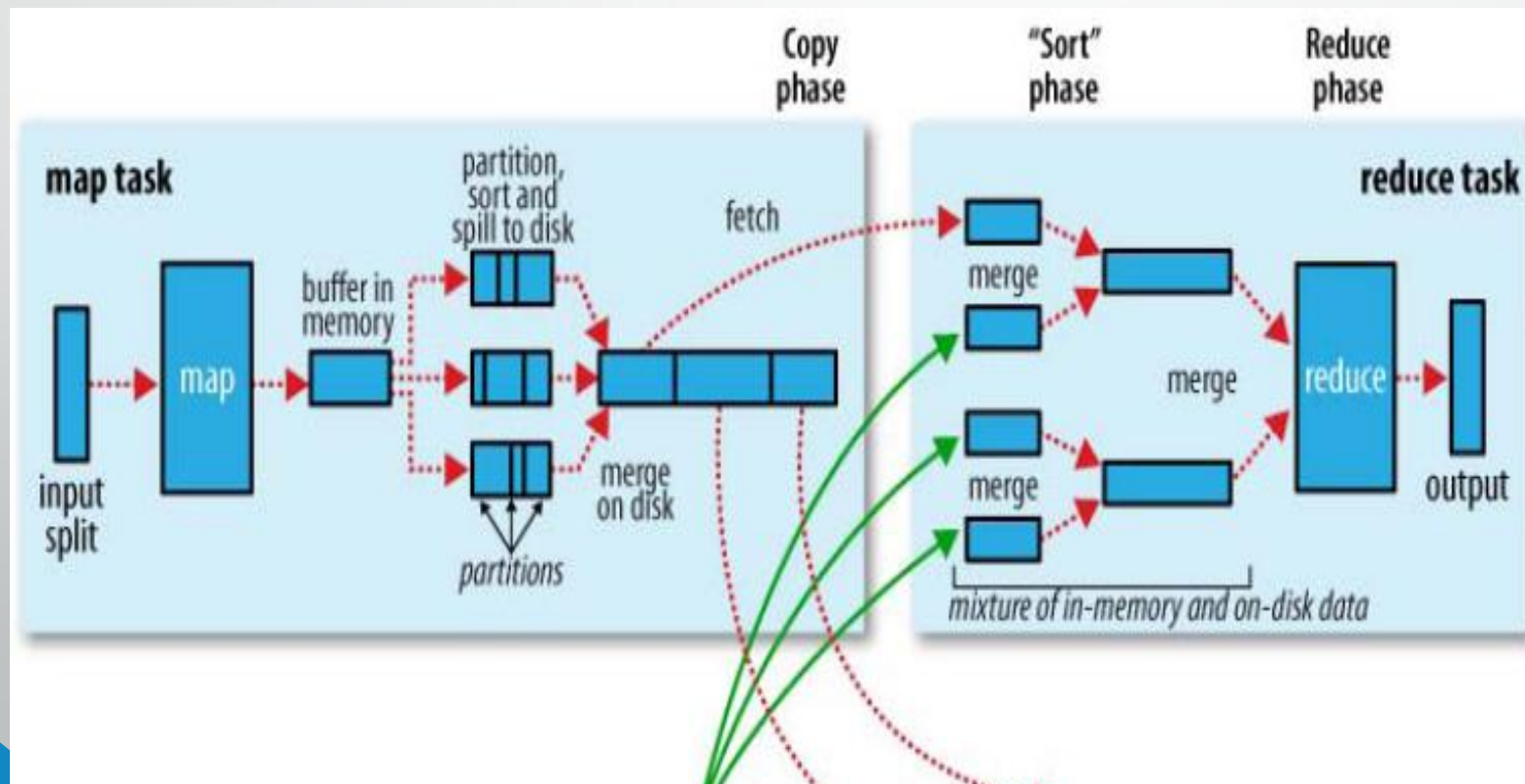
- 作业：用MapReduce实现wordcount功能。

课程大纲

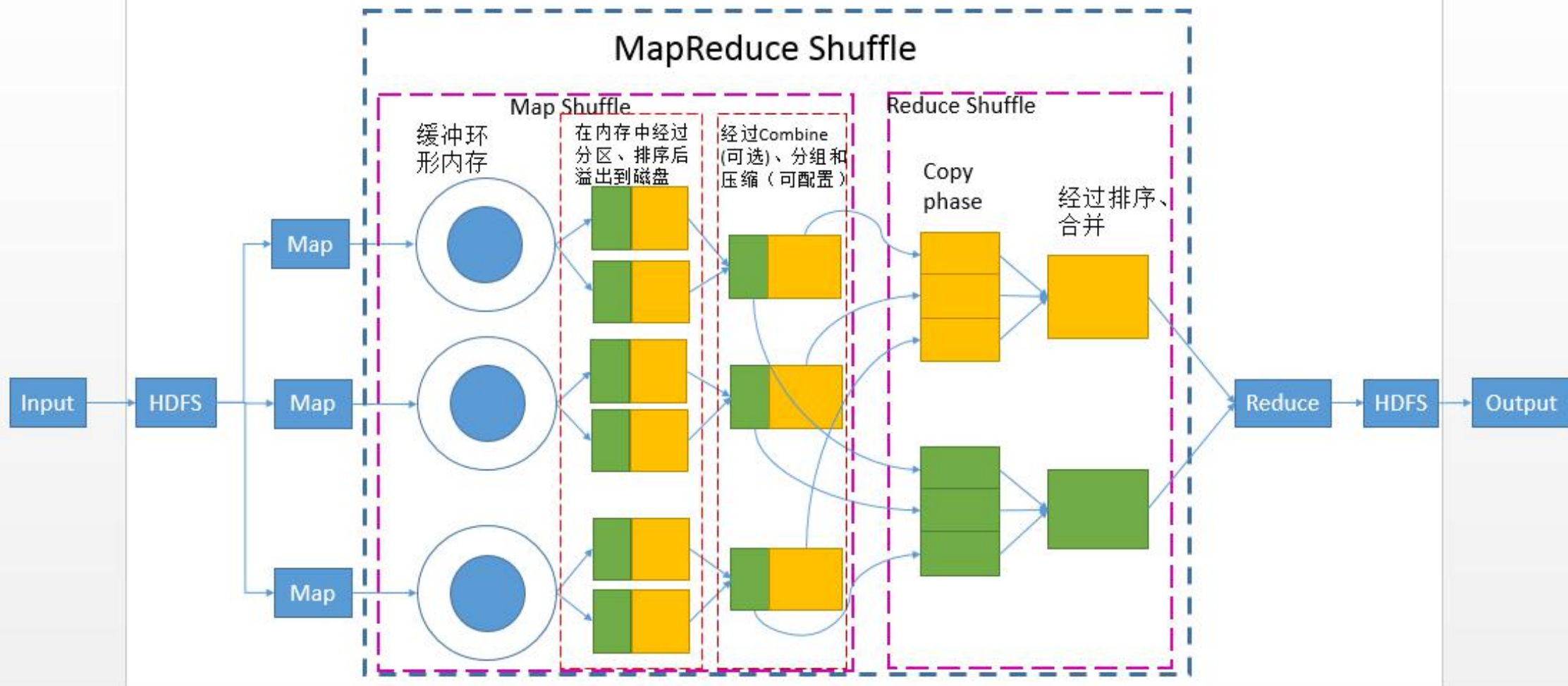
- MapReduce Java API操作流程
- MapReduce 之 Wordcount实例
- **MapReduce Shuffle**
- MapReduce 调优

MapReduce Shuffle

- Shuffle: 洗牌或弄乱。
- Collections.shuffle(List): 随机地打乱参数list里的元素顺序。
- MapReduce里Shuffle: 描述着数据从map task输出到reduce task输入的这段过程。



MapReduce Shuffle



MapReduce Shuffle

- 分区Partitioner
- 排序Sort
- 合并Combiner（可选）
- 压缩Compress（可选）
- 分组Group

- Reduce Task Number
- Map Task 输出压缩
- Shuffle Phase 参数

HADOOP练习： MapReduce Shuffle过程

- 作业： MapReduce Shuffle过程的使用

课程大纲

- MapReduce Java API操作流程
- MapReduce 之 Wordcount实例
- MapReduce 调优

MapReduce 调优

```
<property>
  <name>mapreduce.map.cpu.vcores</name>
  <value>1</value>
  <description>
    The number of virtual cores required for each
  </description>
</property>

<property>
  <name>mapreduce.reduce.cpu.vcores</name>
  <value>1</value>
  <description>
    The number of virtual cores required for each
  </description>
</property>
```

```
<property>
  <name>mapreduce.task.io.sort.factor</name>
  <value>10</value>
  <description>The number of streams to merge at once while sorting
    files. This determines the number of open file handles.</description>
</property>

<property>
  <name>mapreduce.task.io.sort.mb</name>
  <value>100</value>
  <description>The total amount of buffer memory to use while sorting
    files, in megabytes. By default, gives each merge stream 1MB, which
    should minimize seeks.</description>
</property>

<property>
  <name>mapreduce.map.sort.spill.percent</name>
  <value>0.80</value>
  <description>The soft limit in the serialization buffer. Once reached, a
    thread will begin to spill the contents to disk in the background. Note that
    collection will not block if this threshold is exceeded while a spill is
    already in progress, so spills may be larger than this threshold when it is
    set to less than .5</description>
</property>
```