25-hadoop

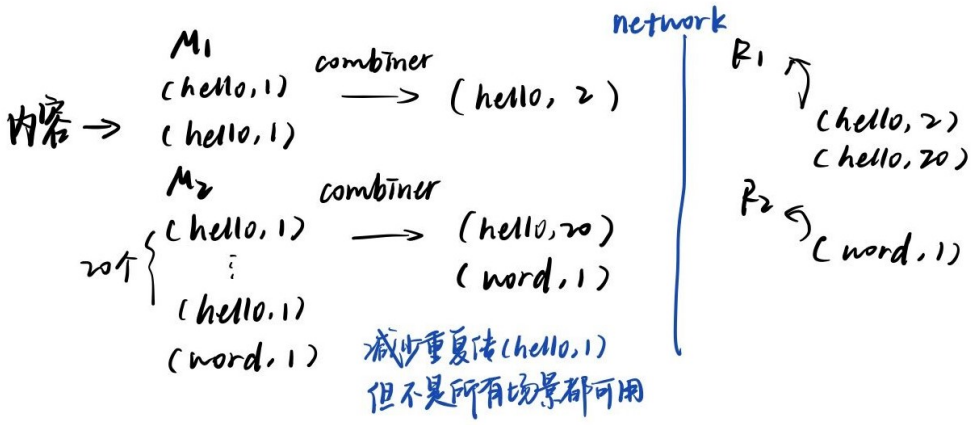
Hadoop的框架最核心的设计就是：HDFS和MapReduce。HDFS为海量的数据提供了存储，则MapReduce为海量的数据提供了计算。

Hadoop：1、Basic Concepts　2、MapReduce　3、YARN

目标：能够针对大数据批处理需求，设计并实现基于MapReduce/YARN的并行处理方案

实例：Word Count

依次获取文件内容→map: 将内容拆分成单词→hadoop中combiner整合结构→reduce: 求和



hadoop像是云上的操作系统

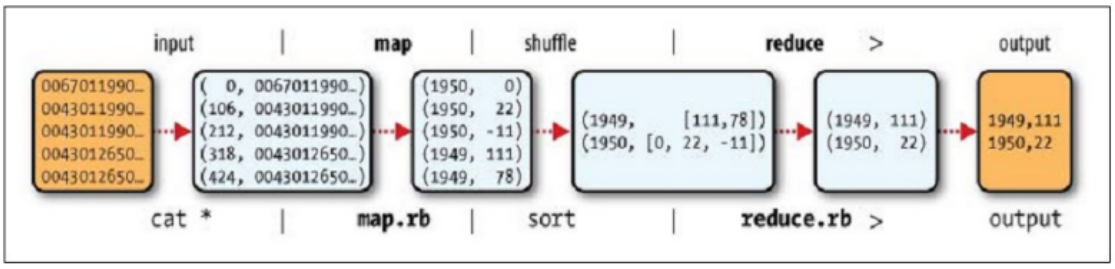
实例：统计每年全球最高温度

```
public class MaxTemperatureReducer
    extends Reducer<Text, IntWritable, Text, IntWritable> {
  @Override
  public void reduce(Text key, Iterable<IntWritable> values,
          Context context)
      throws IOException, InterruptedException {
    int maxValue = Integer.MIN_VALUE;
    for (IntWritable value : values) {
      maxValue = Math.max(maxValue, value.get());
    }
    context.write(key, new IntWritable(maxValue));
  }
}
```

```java
public class MaxTemperatureMapper
    extends Mapper<LongWritable, Text, Text, IntWritable> {
  private static final int MISSING = 9999;
  @Override
  public void map(LongWritable key, Text value, Context context)
      throws IOException, InterruptedException {
    String line = value.toString();
    String year = line.substring(15, 19);
    int airTemperature;
    if (line.charAt(87) == '+') { // parseInt doesn't like leading plus signs
      airTemperature = Integer.parseInt(line.substring(88, 92));
    } else {
      airTemperature = Integer.parseInt(line.substring(87, 92));
    }
    String quality = line.substring(92, 93);
    if (airTemperature != MISSING && quality.matches("[01459]")) {
      context.write(new Text(year), new IntWritable(airTemperature));
    }
  }
}
```

This is the final output:

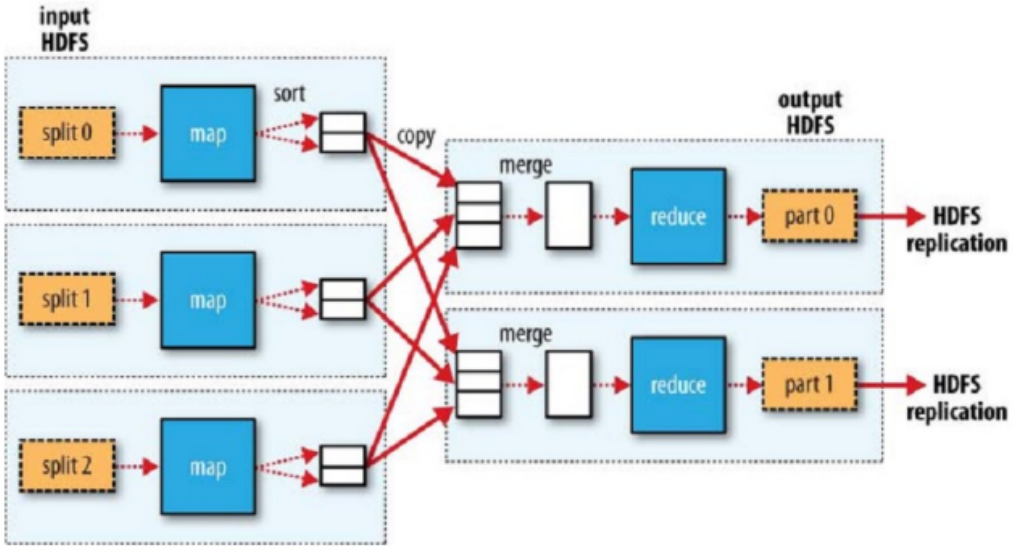
– the maximum global temperature recorded in each year.



map和reduce有几个？谁给了map内容？如何切分那么多数据给不同的map？

这些都是MapReduce框架来帮我们做的

Hadoop可以把将来给map的数据切成几块——split，通常一个split大小等于一个HDFS block的大小（64MB），也可以让我们自己决定。



Job Tracker

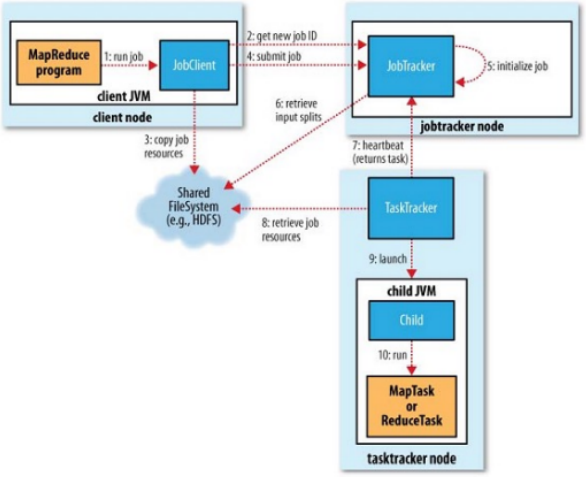
1、看哪些机器空闲着就给它分配个任务

2、跟踪每个job的状态

3、有工作机器挂了→起一个新的机器来干

事情太多了，可能会挂qwq

使用YARN来缓解压力，分成资源管理器和任务管理器，一个任务有一个master



Mapper

数量通常由输入数据的大小来决定，合适的并行量大概是 10-100 maps / 1 node

Reducer How many?

<1: 保守，提供冗余

>1: 快速跑起来

Reducer可能没有

多个Reducer实现输出分区

- The right number of reduces seems to
  - be 0.95 or 1.75 multiplied by (*<no. of nodes> * <no. of maximum containers per node>*).
  - With 0.95 all of the reduces can launch immediately and start transferring map outputs as the maps finish.
  - With 1.75 the faster nodes will finish their first round of reduces and launch a second wave of reduces doing a much better job of load balancing.
- Increasing the number of reduces
  - increases the framework overhead,
  - but increases load balancing and lowers the cost of failures.
- The scaling factors above are slightly less than
  - whole numbers to reserve a few reduce slots in the framework for speculative-tasks and failed tasks.

combiner：减少网络传输，但是有时候不一定能用得上，combiner相当于提前做了一个reduce的工作，减轻了reduce端的压力

partition：分割map每个节点的结果，按照key分别映射给不同的reduce，也是可以自定义的。其实可以理解归类，根据key或value及reduce的数量来决定当前的这对输出数据最终应该交由哪个reduce task处理。划分分区由用户定义的partition函数控制，默认使用哈希函数来划分分区。

自由度体现在：

1、由于map传给reducer的数据可能比较大，可以先压缩再解压，可自行设置压缩大小

2、sort/shuffle的排序顺序是可以自己决定的→自己写compactor

3、每写多少split之后合并一下

补充问答：

1、运行Hadoop集群需要哪些守护进程?

DataNode，NameNode，TaskTracker和JobTracker都是运行Hadoop集群需要的守护进程。

2、Hadoop常见输入格式是什么?

三种广泛使用的输入格式是：

文本输入：Hadoop中的默认输入格式。

Key值：用于纯文本文件

序列：用于依次读取文件

3、RDBMS和Hadoop的主要区别是什么?

RDBMS用于事务性系统存储和处理数据，而Hadoop可以用来存储大量数据。

4、假如Namenode中没有数据会怎么样?

没有数据的Namenode就下能称之为Namenode，通常情况下，Namenode肯定全有故据。

5、当Job Tracker宕掉时，Namenode会发生什么?

当Job Tracker失败时，集群仍然可以正常工作，只要Namenode没问题。