**问题1：在Hadoop中，一个处理文本文件的MapReduce作业，其Map Task数目是如何决定的？**

（1）Hadoop设置的Split与Block的关系来决定。

   （2）输入文件的个数与大小

Map Task的数目是由分片Split的数目决定的，Split是MapReduce中最小的计算单元，一个Split对应一个Map Task，而Split的个数又与Block个数有关。

默认情况下HDFS上一个Block对应一个Split，例如（此处使用Hadoop新版本Block size默认情况为128MB进行说明）：

①输入的一个文件小于128MB，默认情况下则保存在HDFS上的一个Block中，对应一个Split文件，所以将产生一个Map Task。

②如果输入一个129M的文件就对应两个Block，也就对应两个Split，也就产生两个Map Task。

③如果输入为三个小于128M的文件，就对应三个Split，也就产生三个Map Task。

与此同时，用户可自行指定Block与Split的关系，HDSF中的一个Block，一个Split也可以对应多个Block。Split与Block的关系都是一对多的关系。

**问题2：MapReduce中Partitioner的作用是什么？**

Map阶段的输出，需要在Shuffle阶段进行分区和排序等操作。Partitioner的作用实际上就是将Map每个节点的输出结果进行分类，通过key将每一个类的数据映射给不同的Reduce Task处理，所以Partitioner分的份数和Reduce Task的数量是相等的。默认情况下，MapReduce使用的是HashPartitioner，利用key的哈希值进行分区。用户可以继承Partitioner，自定义分区的条件，实现自己的分区逻辑。

**问题3：输入集合如下表所示,按照以下表格的形式，填写Map阶段、Shuffle阶段和Reduce阶段的输出,并简要描述各阶段的操作。**

输入集合如下表所示,按照以下表格的形式

|  |  |
| --- | --- |
| key\_P | key\_C |
| Tom | Jack |
| Jack | Alice |
| Jack | Jesse |

Map阶段：输入的key为行偏移量，value为每行数据，value值用空格分为两个字符串，前者为父母，后者为儿女。其格式为：<行偏移量,父母 儿女>

输出时key为父母则value为child\_儿女，key为儿女则输出parent\_父母。其格式为：<父母,child\_儿女>或<儿女,parent\_父母>。

|  |  |
| --- | --- |
| Map阶段 | |
| 输入 | 输出 |
| <0,Tom Jack>  <8,Jack Alice>  <18,Jack Jesse> | <Tom,child\_Jack>  <Jack,parent\_Tom>  <Jack,child\_Alice>  <Alice,parent\_Jack>  <Jack,child\_Jesse>  <Jesse,parent\_Jack> |

Shuffle阶段：将key相同的键值对进行合并。

|  |  |
| --- | --- |
| Shuffle阶段 | |
| 输入 | 输出 |
| <Tom,child\_Jack>  <Jack,parent\_Tom>  <Jack,child\_Alice>  <Alice,parent\_Jack>  <Jack,child\_Jesse>  <Jesse,parent\_Jack> | <Tom,[child\_Jack]>  <Jack,[parent\_Tom,child\_Alice,child\_Jesse]>  <Alice,[parent\_Jack]>  <Jesse,[parent\_Jack]> |

Reduce阶段：假设一共有n代人(n≥3)，则核心是找到家族中的第k代人（2≤k≤n-1），因为k代人才有可能既有父母又有儿女（不取叶节点和根节点），即在Shuffle阶段输出的结果中value中既有child又有parent（既有前驱又有后继）。扫描所有的key，准备两个临时数组pro和next，扫描value中的内容，parent\_开头的写入pro，child\_开头的写入next，pro和next中的内容两两组合写入context，清空pro和next数组，重复操作，直到遍历完所有key。

在本题中只有三代人，只有Jack是第二代人，Tom写入pro数组，Alice、Jesse写入next数组，两两组合后的结果为<Tom,Alice> <Tom,Jesse>。

|  |  |
| --- | --- |
| Reduce阶段 | |
| 输入 | 输出 |
| <Tom,[child\_Jack]>  <Jack,[parent\_Tom,child\_Alice,child\_Jesse]>  <Alice,[parent\_Jack]>  <Jesse,[parent\_Jack]> | <Tom,Alice>  <Tom,Jesse> |