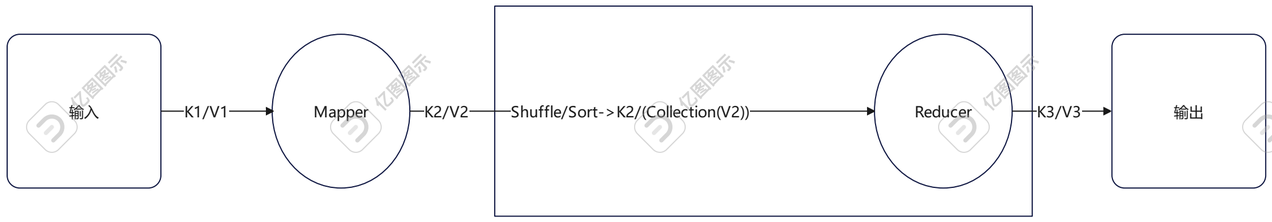
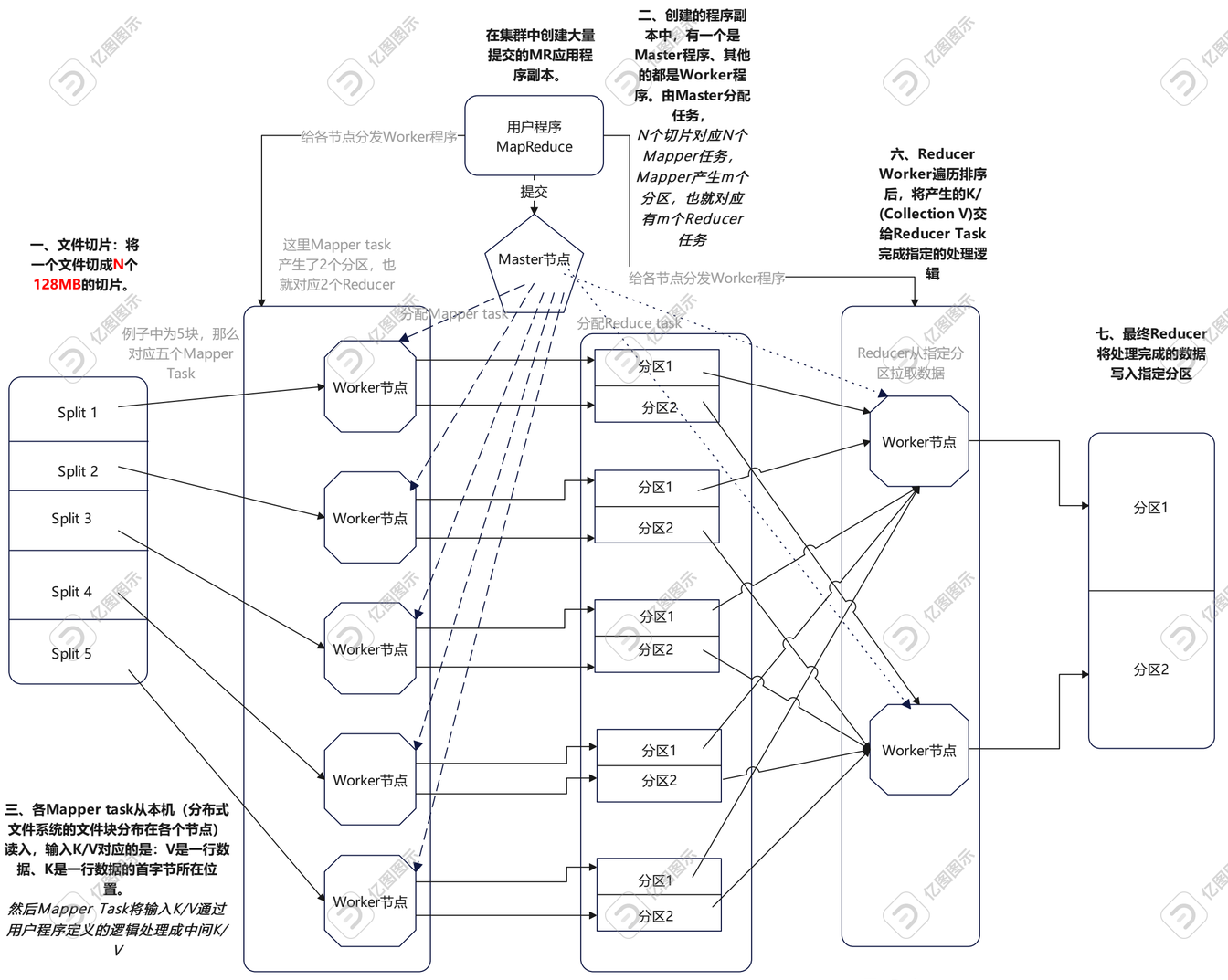
MR数据流


输入
K1/V1
Mapper
K2/V2
Shuffle/Sort->K2/(Collection(V2))
Reducer
K3/V3
输出


在集群中创建大量提交的MR应用程序副本。
二、创建的程序副本中，有一个是Master程序、其他的都是Worker程序。由Master分配任务，
N个切片对应N个Mapper任务，Mapper产生m个分区，也就对应有m个Reducer任务
用户程序
MapReduce
给各节点分发Worker程序
提交
Master节点
一、文件切片：将一个文件切成N个128MB的切片。
这里Mapper task产生了2个分区，也就对应2个Reducer
分配Mapper task
分配Reduce task
例子中为5块，那么对应五个Mapper Task
Split 1
Split 2
Split 3
Split 4
Split 5
Worker节点
Worker节点
Worker节点
Worker节点
Worker节点
分区1
分区2
分区1
分区2
分区1
分区2
分区1
分区2
分区1
分区2
六、Reducer Worker遍历排序后，将产生的K/(Collection V)交给Reducer Task完成指定的处理逻辑
给各节点分发Worker程序
Reducer从指定分区拉取数据
Worker节点
Worker节点
七、最终Reducer将处理完成的数据写入指定分区
分区1
分区2
三、各Mapper task从本机（分布式文件系统的文件块分布在各个节点）读入，输入K/V对应的是：V是一行数据、K是一行数据的首字节所在位置。
然后Mapper Task将输入K/V通过用户程序定义的逻辑处理成中间K/V
四、缓存中的K\V被分为m个分区，数据被周期性的写入磁盘。
缓存被写入磁盘的存储位置信息会被传给Master，master在将这些信息传递给Reducer worker
五、Reducer Worker从Master收到Mapper任务输出分区所存储的位置信息，通过RPC远程读取数据。
读取完中间数据后，将具有相同Key的值聚合在一起。然后进行排序
输入数据不一定是文件，可以实现RecordReader接口来自定义输入数据
Map执行完成后，数据可能出现大量相同的K，为了减少Reducer阶段远程拉取数据产生的网络流量和等待时间，可以指定一个Combiner，提前聚合这些K相同的值
分区的行为可以自定义指定，比如让什么样的Key进入哪个分区。需要继承Partitioner类
远程数据传输涉及到序列化，可以把自定义的类进行序列化，当然要实现Writable接口
二次排序：将Reducer的Sort完成后的K/(Collection V)中的一系列的V进行排序
输出不一定是文件，可以实现RecordWriter接口自定义输出到哪里