**学习日志**

姓名：陈雨舒 学号：1813065 日期：6月24日

**学习内容安排：**

上午：Hadoop的I/O

下午：MapReduce应用开发

**学习反馈：**

Hadoop的I/O

Hadoop会对写入的所有数据计算校验和，并在读取数据时验证校验和。

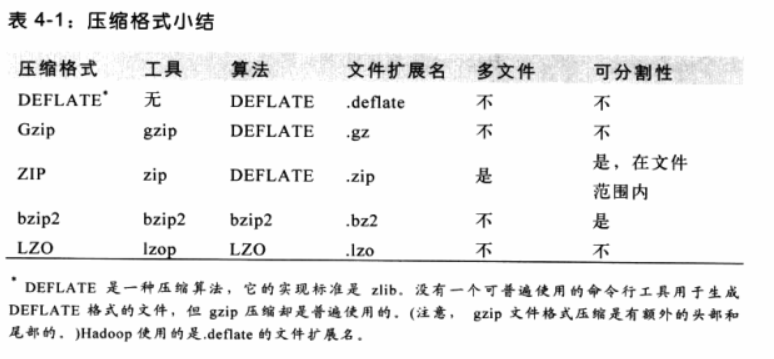
datanode负责在收到数据后存储该数据及其验证校验和。

客户端从datanode读取数据时，也会验证校验和，将它们与datanode中存储的校验和进行比较。Datanode也会在后台线程中运行一个DataBlockScanner定期验证存储在这个datanode上的所有数据块。

HDFS存储这每个数据块的复本，它可以通过数据复本来修复损坏的数据库，得到一个新的、完好无损的复本。基本思路为：客户端读取数据块——>检测到错误——>向namenode报告损坏的数据块及其正在尝试操作的当前的datanode，抛出异常——>namenode标记此数据块复本为已损坏——>namenode安排该数据块的一个复本复制到另一个datanode——>删除已损坏的数据块复本。

2、压缩：

文件压缩两大好处：减少存储文件所需要的磁盘空间，加速数据在网络和磁盘上的传输。

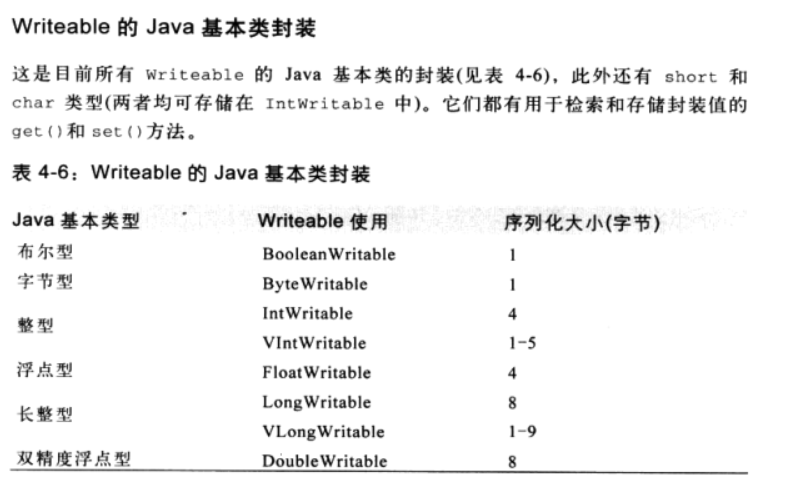


3、序列化：

序列化是指将结构化对象转化为字节流以便在网络上传输或写到磁盘进行永久存储的过程。反序列化是指将字节流转回结构化对象的逆过程。序列化在分布式数据处理的两大领域经常出现：进程间通信和永久存储

在Hadoop中，系统中多个节点上进程间的通信是通过“远程过程调用”（remote procedure call，RPC）实现的。RPC协议将消息序列化成二进制流后发送到远程节点，远程节点接着将二进制流反序列化为原始消息。

通常希望序列化格式比较紧凑（能高效实用存储空间）、快速（以读/写数据的额外开销比较小）、可扩展（以可以透明地读取老格式的数据）且可以互操作（以可以使用不同的语/言读写永久存储的数据）。Hadoop使用自己的序列化格式Writable，它绝对进程、速度快、但不太容易用Java以外的语言进行扩展或使用。Writable是Hadoop的核心，大多数MapReduce程序都是Writable类型的键和值。Writable类对Java基本类型提供封装。所有的封装包含get()和set()两个方法用于读取或存储封装的值。



4、Avro：

Apache Avro是一个独立于编程语言的数据序列化系统，旨在解决Hadoop中Writable类型的不足：缺乏语言的可移植性。Avro通常用JSON来写，数据通常采用二进制格式来编码，但也有其他选择。

5、基于文件的数据结构：

SequenceFiles可作为小文件的容器。HDFS和MapReduce是针对大文件优化的，通过SequenceFiles类型将小文件包装起来可获得更高效的存储和处理。

MapReduce应用开发：

MapReduce编程流程：首先写map函数和reduce函数，使用单元测试确保函数的运行符合预期，然后写一个驱动程序来运行作业（可在本地IDE中用一个小数据集进行测试），最后将通过测试的程序放到集群上运行。

MapReduce的工作流：

JobControl的实例表示一个作业的运行图，可以加入作业的配置，告知JobControl实例作业之间的依赖关系。在一个线程中JobControl将按照依赖顺序执行这些作业。如果一个作业失败，JobControl将不执行与之有依赖关系的后续作业。

Apache Oozie：

Oozie作为服务器运行，客户端提交一个立即或稍后执行的工作流定义到服务器。在Oozie中，工作流是一个由动作（action）节点和控制流节点组成的有向无环图。动作节点执行工作流任务，控制流节点通过构建条件逻辑或并行执行来管理活动之间的工作流执行情况。当工作流结束时，Oozie通过发送一个HTTP的回调向客户端通知工作流的状态。

MapReduce的工作机制：

可通过一个简单的方法调用来运行MapReduce作业：Job对象上的submit()。也可调用waitForCompletion()，它用于提交以前没有提交过的作业，并等待它的完成。submit()方法调用封装了大量的处理细节。

MapReduce 的输入输出模型：  
MapReduce 中，reduce函数的输入类型必须与map 函数的输出类型一致

MapReduce 中的常用设置：

输入数据类型由输入格式(InputFormat)设置  
map 输出的key 类型通过setMapOutputKeyClass 设置  
MapReduce 输入格式：  
MapReduce 处理的输入文件一般存储在HDFS上,这些输入文件的格式多种多样，比如基于行的日志文件、二进制文件等等，而且有些文件非常之大。因此，MapReduce 定义了一些接口来读取这些文件数据。

InputFormat 接口  
InputFormat 能够从一个job中得到一个split 集合，再为这个split集合配上一个合适的RecordReader来读取每个split 中的数据。

MapReduce 输出格式：  
Hadoop 默认情况下只有一个Reduce，即输出只有一个文件。输出文件的个数有Reduce的个数一致。

OutputFormat 接口：  
OutputFormat 主要用于描述数据的格式，它能够将用户提供的key/value 对写入到特定格式的文件中。

Combiner ：  
Hadoop 框架使用Mapper 将数据处理成一个<key,value> 键值对，再在网络空间中对其进行整理(shuffile)，然后使用Reducer 处理数据并进行最终输出。

Partitioner:  
在进行MapReduce 计算时，有时候需要将最终的输出数据分到不同的文件中，比如按照省份、性别等等来进行划分。如果想要得到多个文件，就意味着有同等数量的Reducer 任务在运行。Reducer任务的数据来自于Mapper 任务，也就是说Mapper 任务要划分数据，将不同的数据分配给给不同的Reducer 运行，Mapper 任务划分数据的过程称为Partition,负责划分数据的类为Partitioner。  
自定义RecordReader：  
RecordReader 表示以怎样的方式从分片中读取一条记录，每读取一条记录都要调用一次RecordReader 类，系统默认的RocordReader 是 LineRecorderReader,它是TextInputFormat 对应的RecordReader。SequenceFileInputFormat 对应的RecordReader 是SequenceFileRecordReader。

LineRecordReader 以每行的偏移量作为读入Map 的Key,每行读入的内容作为Map 的Value。  
有时候我们需要自定义RecordReader以我们需要的方式去读取记录，比如以文件名或者行号去读取记录。

未掌握知识：

初步了解了Hadoop的I/O和MapReduce应用开发的相关知识，但有些知识点没有应用到实际中，对集群还不是很了解。

学习心得：

通过一天的学习，初步了解了Hadoop的I/O和MapReduce应用开发的相关知识，需要进一步在实践中熟练，进一步了解Hadoop。