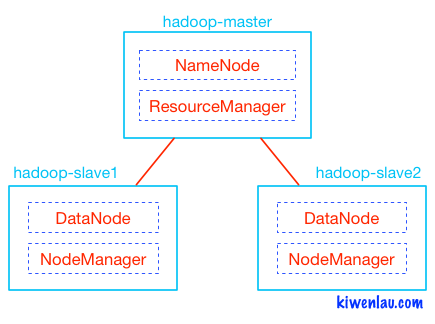
大数据四门语言R、Python、Scala、Java 比较 <http://www.sohu.com/a/72778066_355129> 。Maven，SBT，Gradle以及自动安装依赖关系都是针对Java和Scala的。Python一般只需要PiP。由Java和Scala应用程序构建一个JAR需要一段时间，对于许多Hadoop来说，JVM语言有一定的优势。您可以在JVM上运行JPython，但我真的没有看到用于大数据，Spark或机器学习。

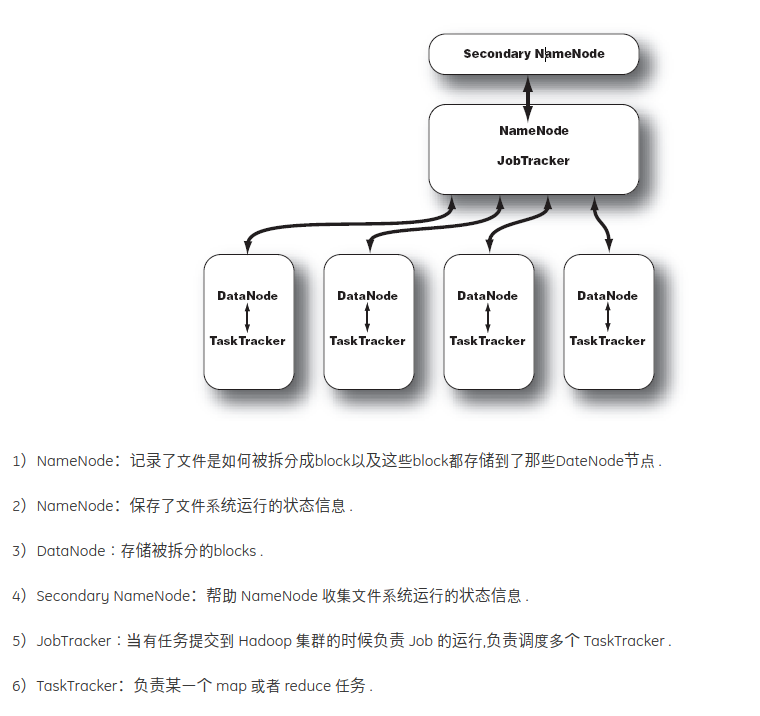
数据来源主要分三种情况，一是，自有平台产生，像腾讯、淘宝、华西医院。二是，靠网络采集系统，像百度、谷歌。三是，像第三方数据供应商提供，如运营商数据、开源大数据共享集合等。大数据即对大量数据中隐藏信息进行挖掘，比如zz做过通过某个身体部位的骨骼图片来识别年龄。

大的信息检索系统分布式，老师说并不是将数据全部都装入内存，放不下，所以用到了外存操作与外存排序。但是我认为从百度与google的检索速度来看应该不是这样的。

docker搭建Hadoop集群：https://github.com/kiwenlau/hadoop-cluster-docker

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原本来自于谷歌一款名为MapReduce的编程模型包。谷歌的MapReduce框架可以把一个应用程序分解为许多并行计算指令，跨大量的计算节点运行非常巨大的数据集。Hadoop带有用[Java](https://baike.baidu.com/item/Java" \t "/home/vega/Documents\\x/_blank)语言编写的框架，因此运行在 Linux 生产平台上是非常理想的。Hadoop 上的[应用程序](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F" \t "/home/vega/Documents\\x/_blank)也可以使用其他语言编写，比如 [C++](https://baike.baidu.com/item/C++" \t "/home/vega/Documents\\x/_blank)。通常有较高的延迟并且在作业提交与调度的时候需要大量开销。





Hadoop推荐使用Oracle JDK，而不是完全开源的OpenJDK，避免不兼容问题。

各个端口说明： <https://blog.csdn.net/cnhk1225/article/details/52281029>

Hadoop中的变量类型：<https://www.cnblogs.com/codeOfLife/p/5400427.html>

NullWritable.get() 就会获取一个NullWritable对象。

Java里面的基本类型与包装类默认没有实现类的Serializbale借口，所以不能被序列化，于是Hadoop需要使用自己定义的支持序列化的类型,org.apache.hadoop.io包中的Writable接口是Hadoop序列化格式的实现。

只有可以将数据独立分离计算与合并的任务才可以用hadoop来搞定，即可分割。但大量的实战证明，绝大多数的计算任务都可以通过合理的设计化分成可以划分成可以hadoop处理的任务。但像gzip压缩包解压缩这样的操作，一般认为是无法用hadoop来并行计算的，因为gzip是不可分割的，但像lzo,bz2等可分割的压缩格式，均可以被hadoop处理。

像实时查询、实时计算对于hadoop是不适合的。为此hadoop也开发如hive、hbase等为解决这一问题作出努力，但在实时性方面还是很不适用的，像最近兴趣的spark、storm等，可以作为大数据实时处理的利器，与hadoop互补使用。

对于分布式软件，故障的监测与快速修复是必须的。

HDFS Hadpoop Distribute File System :

被设计成通用硬件上的分布式文件系统（java写的，基于jvm）。HDFS是一个主从结构，一个HDFS集群是由一个名字节点，它是一个管理文件命名空间和调节客户端访问文件的主服务器，当然还有一些数据节点，通常是一个节点一个机器，它来管理对应节点的存储。HDFS对外开放文件命名空间并允许用户数据以文件形式存储。一个文件被分块存储在多个数据节点上。负责海量数据的存储，集群中的角色主要有 NameNode / DataNode。

DistributedFileSystem通过RPC（Remote Procedure Call）调用询问NameNode。

三种node：namenode、datanode、clientnode（这个node与数据存储无关，但可以发出mapreduce job等人为操作）

一个假定就是迁移计算到离数据更近的位置比将数据移动到程序运行更近的位置要更好。HDFS提供了接口，来让程序将自己移动到离[数据存储](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%AD%98%E5%82%A8" \t "/home/vega/Documents\\x/_blank)更近的位置。

机器之间的的通讯基于TCP/IP，运行上的数据量大小GB、TB级别。

分布式文件系统的容错只能通过对文件块的复制。datanode直接进行复制就好了，没有必要都与clientnode建立连接。

文件系统的实现需要选择合适的读写算法，文件树算法（避免环的产生）、文件操作算法（比如删除就是对树的处理）。

文件系统相关命令见hdfs.docx，我建议通过alias重命名”hdfs dfs”。

//hdfs dfs操作

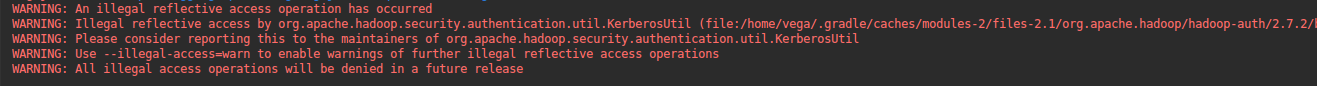
Configuration configuration = new Configuration();

FileSystem fileSystem = FileSystem.get(URI.create("hdfs://192.168.1.120:9000"), configuration, "root");

//9000端口与hdfs沟通

FileInputFormat.addInputPath(job, new Path("hdfs://192.168.1.111:9000/user/input/wc/"));

hadoop最新版但是也应该用JDK8而不是JDK9、 10，否则warning

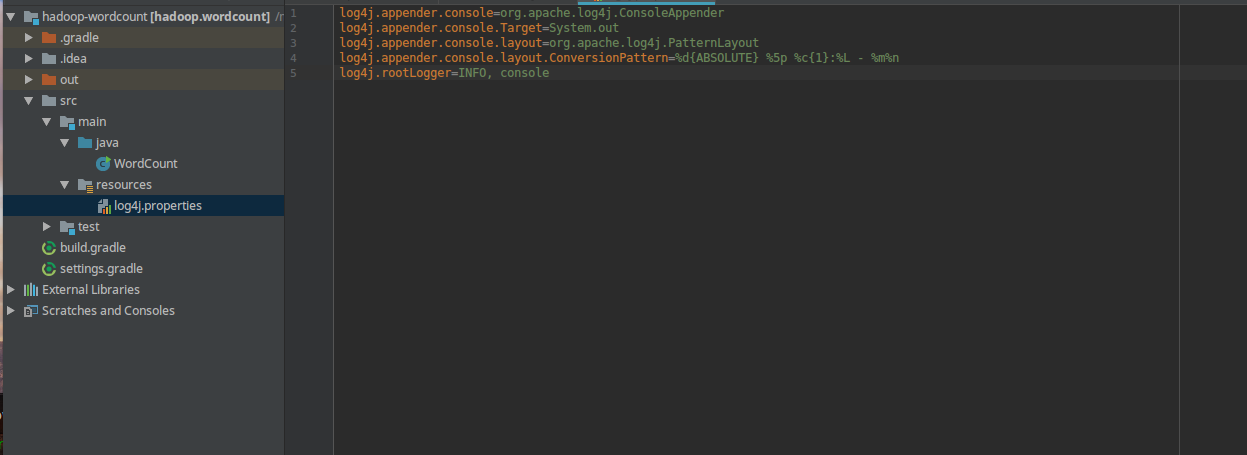


log4j系列错误:

log4j:WARN No appenders could be found for logger (org.apache.hadoop.metrics2.lib.MutableMetricsFactory).  
log4j:WARN Please initialize the log4j system properly.  
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.

这种情况一般是由于log4j这个日志信息打印模块的配置信息没有给出造成的。在resources中创建一个log4j.properties:

log4j.appender.console=org.apache.log4j.ConsoleAppender log4j.appender.console.Target=System.out log4j.appender.console.layout=org.apache.log4j.PatternLayout log4j.appender.console.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n log4j.rootLogger=INFO, console



本机为vega用户没有权限对docker hadoop的hdfs进行写操作

1. 关闭dfs的权限系统。在hdfs的配置文件中，将dfs.permissions修改为False。
2. hadoop fs -chmod 777 /user/hadoop
3. 在系统的环境变量或java JVM变量里面添加对应的HADOOP\_USER\_NAME

原理http://www.huqiwen.com/2013/07/18/hdfs-permission-denied/

GFS ： Google File System

YARN：

负责海量数据运算时的资源调度，集群中的角色主要有 ResourceManager /NodeManager

jps命令可以在datanode上看到yarnchild。

MRAppMaster是MapReduce的ApplicationMaster实现，它使得MapReduce可以直接运行在YARN上，老师在datanode上看到的不是yarnchild而死MRAppMaster。

运行一个任务之后（比如mapreduce），就会在HDFS的/tmp中产生一些记录文件：

两个优化思想： 减少网络输出量，减少计算量，尽可能使用内存

MapReduce： map -->shuffle-->reduce

mapreduce只是hadoop的一种Application Type，最后结果是按照key的从小到大顺序输出，虽然两次排序都是可以自己定制的，但是如果有需要最后按照value的排序来输出，最好不要用MapReduce，或者https://blog.csdn.net/ymybxx/article/details/78581235 。

hadoop的map/reduce都是以进程为单位进行计算的，reduce设置的过多会导致reduce的初始化与销毁浪费时间，从而影响整个任务的效率。要根据输出数据量来计算reduce的数量。

job：运行作业的客户端通过调用getSplits()计算分片，然后将他们发送到jobtracker。jobtracker使用其存储位置信息来调度map任务从而在tasktracker上处理这些分片数据。

map：map函数接受一个键值对（key-value pair），产生一组中间键值对（由输入的value得到许多value）。map任务把经过规则处理后的分片（有个默认规则是把每一行文本内容解析成键值对，这里的“键”是每一行的起始位置(单位是字节)，“值”是本行的文本内容）传给InputFormat的getRecordReader（）方法来获得这个分片的RecordReader。RecordReader就像是记录上的迭代器，map任务通过调用mapper的run（）方法用一个RecordReader来生成记录的键/值对，进而将该键/值对传给mapper的map方法作为输入。

Shuffle：

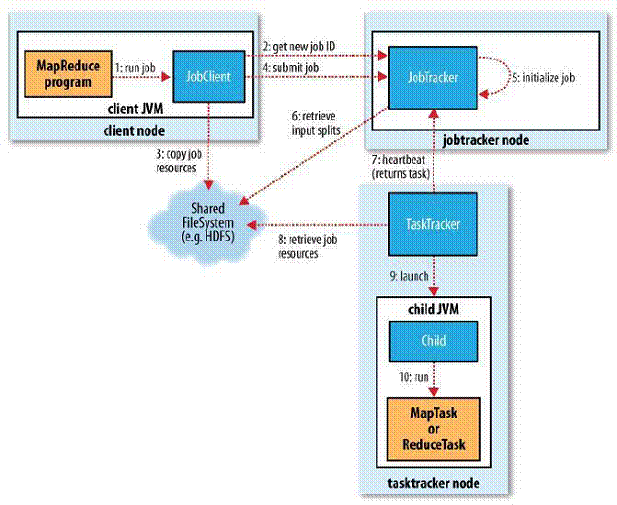
hadoop提供combiner来合并map输出的零散小数据，从而减小网络传输，提高整体的效率。还有减少对磁盘的IO。Combine在整个过程中会多次触发。

mapper方法的输出按照partition规则，讲输出写入不同的环形缓冲区中（默认对key hash后再以reduce task数量取模。默认的取模方式只是为了平均reduce的处理能力）（桶排序思想）。输出的过程中存在按照序列化的key排序与combine（按照reduce规则合并同key数据的value）。当内存缓冲区快满的时候就会溢写到磁盘，如果内存缓冲区已经满了就会阻塞map任务，持续对磁盘的写入。最后所有的输出都会写入磁盘，之前溢写是流写入，现在是块写入。

Task Tracker会不断询问Map Task是否完成，如果某台Task Tracker上的Map Task已经完成，就会把这多个临时文件合并，即做merge操作，注意，这里的merge操作只是简单的合并，如果没有在该处设置Combiner，是不会对相同key进行压缩的，所以可能会有相同的key出现。merge操作就是对于同样的key，其value变为list，把多个value放在list中。这种key/value的形式就是reduce的输入数据格式。

reduce函数：接受一个键，以及相关的一组值，将这组值进行合并产生一组规模更小的值（通常只有一个或零个值）。

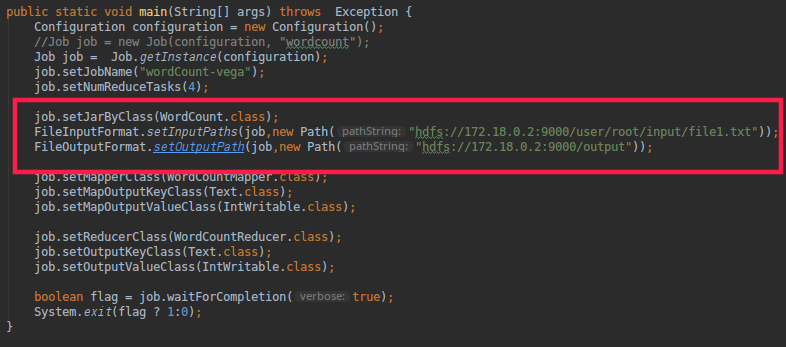
在统计词频的例子里，map函数接受的键是文件名，值是文件的内容，map逐个遍历单词，每遇到一个单词w，就产生一个中间键值对<w, "1">，这表示单词w咱又找到了一个；MapReduce将键相同（都是单词w）的键值对传给reduce函数，这样reduce函数接受的键就是单词w，值是一串"1"（最基本的实现是这样，但可以优化），个数等于键为w的键值对的个数，然后将这些“1”累加就得到单词w的出现次数。最后这些单词的出现次数会被写到用户定义的位置，存储在底层的分布式存储系统（GFS或HDFS）。



在hadoop运行程序有两种方法：

1. eclipse连接编译软件与hadoop

idea上我是在程序中制定了node节点的IP与端口



1. 打包成jar包

hadoop jar xxx.jar Packagename.Classname 程序main函数参数

Spark:

Spark，拥有Hadoop MapReduce所具有的优点；但不同于MapReduce的是——Job中间输出结果可以保存在内存中，从而不再需要读写HDFS，因此Spark能更好地适用于数据挖掘与机器学习等需要迭代的MapReduce的算法。与Hadoop的MapReduce相比，Spark基于内存的运算要快100倍以上，基于硬盘的运算也要快10倍以上。Spark实现了高效的DAG执行引擎，可以通过基于内存来高效处理数据流。Spark可以使用Hadoop的YARN和Apache Mesos作为它的资源管理和调度器。

Shark = Spark+HiveQL 被SparkSQL取代

SparkR = Spark + R

SparkMaster：

WebUI: 8080

URL:          spark://supermap:7077  
 REST URL: spark://supermap:6066(cluster mode)

数据仓库Data Warehouse， 研究与解决从数据库中获取信息的问题，帮助决策者从大量数据中分析出有用的信息，所以说是面向主题的。

Hive ： hadoop的一个数据仓库工具，将结构化的数据文件映射为一张数据表，并提供简单的sql查询功能，可以将sql语句转换为MapReduce任务运行。Hive 不支持对数据的改写和添加，所有的数据都是在加载的时候确定的，数据仓库的作用就是分析数据。支持web UI、命令行操作。