

****Block数据块;****

****基本存储单位，一般大小为64M（配置大的块主要是因为：1）减少搜寻时间，一般硬盘传输速率比寻道时间要快，大的块可以减少寻道时间；2）减少管理块的数据开销，每个块都需要在NameNode上有对应的记录；3）对数据块进行读写，减少建立网络的连接成本）****

****一个大文件会被拆分成一个个的块，然后存储于不同的机器。如果一个文件少于Block大小，那么实际占用的空间为其文件的大小****

****基本的读写单位，类似于磁盘的页，每次都是读写一个块****

****每个块都会被复制到多台机器，默认复制3份****

****NameNode****

****存储文件的metadata，运行时所有数据都保存到内存，整个HDFS可存储的文件数受限于NameNode的内存大小****

****一个Block在NameNode中对应一条记录（一般一个block占用150字节），如果是大量的小文件，会消耗大量内存。同时map task的数量是由splits来决定的，所以用MapReduce处理大量的小文件时，就会产生过多的map task，线程管理开销将会增加作业时间。处理大量小文件的速度远远小于处理同等大小的大文件的速度。因此Hadoop建议存储大文件****

****数据会定时保存到本地磁盘，但不保存block的位置信息，而是由DataNode注册时上报和运行时维护（NameNode中与DataNode相关的信息并不保存到NameNode的文件系统中，而是NameNode每次重启后，动态重建）****

****NameNode失效则整个HDFS都失效了，所以要保证NameNode的可用性****

****Secondary NameNode****

****定时与NameNode进行同步（定期合并文件系统镜像和编辑日志，然后把合并后的传给NameNode，替换其镜像，并清空编辑日志，类似于CheckPoint机制），但NameNode失效后仍需要手工将其设置成主机****

****DataNode****

****保存具体的block数据****

****负责数据的读写操作和复制操作****

****DataNode启动时会向NameNode报告当前存储的数据块信息，后续也会定时报告修改信息****

****DataNode之间会进行通信，复制数据块，保证数据的冗余性****