# HDFS相关

## 定义

1. hdfs是分布式的文件系统
2. 使用场景：适合一次写入，多次读出的场景，并且不支持文件修改
3. 不提供文件的寻址功能
4. 原因：hdfs上存储，是以block为基本单位进行存储（最大128MB）

## 特点（优缺点）

1. 高容错性
   1. 数据保存多个副本
   2. 某一个副本丢失后，可以自动恢复
2. 适合处理大数据
3. 可以构建在廉价的机器上
4. 不适合低延迟数据访问
5. 无法高效对大量小文件进行存储
   1. 小文件，会占用Namenode内存，用于存储文件目录和块信息
   2. 内存利用率偏低
6. 不支持并发写入，文件的随机修改

## block大小2.7版本128MB，老版本64MB

1. 取决于参数dfs.blocksize
2. 原因
   1. 最佳传输损耗原理
   2. 理论：在一次传输中，寻址时间占总传输时间的1%时，本次传输的损耗最小。
   3. 普通磁盘写速度是100MB/s,寻址时间是1ms
   4. 同理，如果是固态硬盘300MB/s,block=256MB
   5. 同理，如果是固态硬盘500MB/s,block=512MB
3. 块大小需要合适调节
   1. 不能太大
   2. 不能太小
4. hdfs shell
   1. http://hadoop.apache.org/docs/r2.7.2/hadoop-project-dist/hadoop-hdfs/HDFSCommands.html
5. Java客户端

|  |
| --- |
| private FileSystem fs;  private Configuration conf = new Configuration();  @Before  public void init() throws IOException, URISyntaxException {  conf.set("fs.hdfs.impl", "org.apache.hadoop.hdfs.DistributedFileSystem");  //创建一个客户端对象  fs=FileSystem.get(new URI("hdfs://h11:9000"),conf);  }  @After  public void close() throws IOException {  if (fs !=null) {  fs.close();  }  }      @Test  public void fun() throws URISyntaxException, IOException, InterruptedException {    System.out.println(fs.getCanonicalServiceName());    fs.mkdirs(new Path("/vova\_j"));  } |