

## Hadoop文件系统元数据fsimage和编辑日志edits

在《[Hadoop NameNode元数据相关文件目录解析](#)》

文章中提到NameNode的\$dfs.namenode.name.dir/current/文件夹的几个文件：

```
current/  
|-- VERSION  
|-- edits_*  
|-- fsimage_0000000000008547077  
|-- fsimage_0000000000008547077.md5  
`-- seen_txid
```

其中存在大量的以edits开头的文件和少量的以fsimage开头的文件。那么这两种文件到底是什么，有什么用？下面对这两中类型的文件进行详解。在进入下面的主题之前先来搞清楚edits和fsimage文件的概念：

(1)、fsimage文件其实是Hadoop文件系统元数据的一个永久性的检查点，其中包含Hadoop文件系统的所有目录和文件idnode的序列化信息；

(2)、edits文件存放的是Hadoop文件系统的所有更新操作的路径，文件系统客户端执行的所以写操作首先会被记录到edits文件中。

fsimage和edits文件都是经过序列化的，在NameNode启动的时候，它会将fsimage文件中的内容加载到内存中，之后再执行edits文件中的各项操作，使得内存中的元数据和实际的同步，存在内存中的元数据支持客户端的读操作。

NameNode起来之后，HDFS中的更新操作会重新写到edits文件中，因为fsimage文件一般都很大（GB级别的很常见），如果所有的更新操作都往fsimage文件中添加，这样会导致系统运行的十分缓慢，但是如果往edits文件里面写就不会这样，每次执行写操作之后，且在向客户端发送成功代码之前，edits文件都需要同步更新。如果一个文件比较大，使得写操作需要向多台机器进行操作，只有当所有的写操作都执行完成之后，写操作才会返回成功，这样的好处是任何的操作都不会因为机器的故障而导致元数据的不同步。

fsimage包含Hadoop文件系统的所有目录和文件idnode的序列化信息；对于文件来说，包含的信息有修改时间、访问时间、块大小和组成一个文件块信息等；而对于目录来说，包含的信息主要有修改时间、访问控制权限等信息。fsimage并不包含DataNode的信息，而是包含DataNode上块的映射信息，并存放内存中，当一个新的DataNode加入到集群中，DataNode都会向NameNode提供块的信息，而NameNode会定期的“索取”块的信息，以使得NameNode拥有最新的块映射。因为fsimage包含Hadoop文件系统的所有目录和文件idnode的序列化信息，所以如果fsimage丢失或者损坏了，那么即使DataNode上有块的数据，但是我们没有文件到块的映射关系，我们也无法用DataNode上的数据！所以定期及时的备份fsimage和edits文件非常重要！

在前面我们也提到，文件系统客户端执行的所以写操作首先会被记录到edits文件中，那么久而久之，edits会非常的大，而NameNode在重启的时候需要执行edits文件中的各项操作，那么这样会导致NameNode启动的时候非常长！在下篇文章中我会谈到在Hadoop 1.x版本和Hadoop 2.x版本是怎么处理edits文件和fsimage文件的。

本博客文章除特别声明，全部都是原创！  
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。  
本文链接：[【】（）](#)