Hadoop 2.0 Trash

在DistributedRaidFileSystem中，包含了方法undelete(Path f,String userName)，这个方法先调用了DistributedFileSystem.undelete()恢复源文件，然后通过通过Codec.parityDirectory获取校验文件所在的目录，然后恢复校验文件。

但是在Hadoop 2.0中，没有undelete方法，在类FileSystem中增加了undelete，DistributedFileSystem继承了FileSystem，并实现了undelete方法，实现代码如下所示：

*public boolean undelete(Path f, String userName) throws IOException {*

*Trash trash = new Trash(this, getConf(), userName);*

*return trash.moveFromTrash(f);*

*}*

从代码中可以看出，使用了类Trash来实现文件的恢复。

在HDFS中，删除文件时，不会真正的删除，而是放入回收站/.trash中，回收站里的文件可以快速恢复，Hadoop回收站trash功能默认是关闭的，需要在core-site.xml中手动开启，配置fs.trash.interval。

HDFS会为每个用户创建一个回收站目录，/user/用户名/.Trash，每个被通过shell删除的文件或目录，在系统回收站中都有一个时间周期，过了这个时间，就彻底被删除。Namenode通过后台线程TrashPolicyDefault.Emptier来定时清空回收站的文件。但是通过API来删除文件或目录时，并不会放入回收站Trash，而是自己实现相关的回收站逻辑。

## 1 Trash

文件的删除，其实就是将文件先放到回收站目录中，定期通过API来删除。

成员变量:

TrashPolicy trashPolicy配置的trash policy实例。

成员方法：

moveFromTrash():从trash中将文件恢复到原目录

moveToTrash():删除文件至trash中

getEmptier():获取线程，该线程用于定期清理回收站内的文件

## 2 TrashPolicyDefault

TrashPolicy是抽象类，其中方法getInstance(conf,fs,home)，根据fs.trash.classname的配置选择实际TrashPolicy类。TrashPolicyDefault是默认使用的类

成员变量：Path current:当前路径

Path homesParent: 主路径

long empierInterval: 清理时间周期

内部类：Emptier，

run()：检查目录，并使用trash.deleteCheckPoint()删除回收站中的文件，checkPoint()用于标记文件时间，用于将超过存留时间的文件删除。

文件的删除操作通过FileSystem.delete(path,true)来完成

## 3 moveFromTrash()

但是在hadoop 2.0中是没有这个方法的，不支持文件从回收站中恢复，但是在FB 0.20中实现了方法moveFromTrash，该方法用于Undelete()恢复删除的文件。

注意：Trash的构造函数Trash(fs,conf,userName)，这个构造方法是没有的，没有参数userName，在调试过程中要注意这个问题。