

三种恢复 HDFS 上删除文件的方法

我们每天都可能会操作 HDFS 上的文件，这就很难避免误操作，比如比较严重的误操作就是删除文件。本文针对这个问题提供了三种恢复误删除文件的方法，希望对大家的日常运维有所帮助。



如果想及时了解 Spark、Hadoop 或者 Hbase 相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

通过垃圾箱恢复

HDFS 为我们提供了垃圾箱的功能，也就是当我们执行 `hadoop fs -rmr xxx` 命令之后，文件并不是马上被删除，而是会被移动到执行这个操作用户的 `.Trash` 目录下，等到一定的时间后才会执行真正的删除操作。看下下面的例子：

```
$ sudo -u iteblog hadoop fs -rmr /user/iteblog/test.txt
Moved: 'hdfs://iteblogcluster/user/iteblog/test.txt' to trash at: hdfs://iteblogcluster/user/iteblog/.Trash/Current
```

```
$ sudo -u iteblog hadoop fs -ls /user/iteblog/.Trash/Current/user/iteblog
-rw-r--r--  3 iteblog iteblog 103 2017-05-15 17:24 /user/iteblog/.Trash/Current/user/iteblog/test.txt
```

```
$ sudo -u iteblog hadoop fs -mv /user/iteblog/.Trash/Current/user/iteblog/test.txt /user/iteblog/
```

```
$ sudo -u iteblog hadoop fs -ls /user/iteblog/test.txt  
-rw-r--r--  3 iteblog iteblog 103 2017-05-15 17:24 test.txt
```

从上面的例子中可以看出，我们删了 test.txt 文件之后，文件被移到 /user/iteblog/.Trash/Current/user/iteblog/test.txt 路径下，如果这个操作属于误操作，那么我们可以到回收站找回这个文件并直接 mv 回原来的目录即可恢复之前的数据。不过这个功能的前提是要求我们启用 fs.trash.interval 参数，默认是 0 代表不启用垃圾箱功能。

```
<property>  
  <name>fs.trash.interval</name>  
  <value>1440</value>  
  <description>  
    Number of minutes after which the checkpoint gets deleted. If zero, the trash feature is disabled. This option may be configured both on the server and the client. If trash is disabled server side then the client side configuration is checked. If trash is enabled on the server side then the value configured on the server is used and the client configuration value is ignored.  
  </description>  
</property>
```

上面的配置是说，文件被删除会保留到 .Trash 目录下一天，超过这个时间被删除的文件就会真正被删除。所以为了误删除操作，强烈建议开启 HDFS 回收站功能。

通过快照恢复

Hadoop 从 2.1.0 版本开始提供了 HDFS 快照 (SnapShot) 功能。一个快照是一个全部文件系统、或者某个目录在某一时刻的镜像。利用快照可以防止用户错误操作，管理员可以通过以滚动的方式周期性设置一个只读的快照，这样就可以在文件系统上有若干份只读快照。如果用户意外地删除了一个文件，就可以使用包含该文件的最新只读快照来进行恢复。下面我们来实操说明如何利用快照恢复误删除的文件：

创建目录和文件

```
$ sudo -u bizdata hadoop fs -mkdir /user/iteblog/important/  
$ echo "important data" | sudo -u iteblog hadoop fs -put - /user/iteblog/important/important-file.txt  
$ sudo -u iteblog hadoop fs -cat /user/iteblog/important/important-file.txt  
important data
```

上面我们创建了 /user/iteblog/important/ 目录，里面有一个文件 important-file.txt，假设这个文件是非常重要的。

创建快照

```
$ sudo -u iteblog hadoop dfsadmin -allowSnapshot /user/iteblog/important
$ sudo -u iteblog hadoop fs -createSnapshot /user/iteblog/important important-snapshot
```

现在我们已经为 important 目录创建了快照，名称为 important-snapshot。

误删除操作

因为开启了快照功能，我们无法删除已经创建快照的目录（/user/iteblog/important），但是我们依然可以删除这个目录下的文件；

```
$ sudo -u iteblog hadoop fs -rm -r /user/iteblog/important/important-file.txt
```

现在这个重要的文件被我们误删除了！

恢复文件

别急，因为我们开启了快照，所有我们可以从快照中恢复这个文件，步骤如下：

```
$ sudo -u iteblog hadoop fs -ls /user/iteblog/important/.snapshot/
$ sudo -u iteblog hadoop fs -cp /user/iteblog/important/.snapshot/important-snapshot/important-file.txt /user/iteblog/important/
$ sudo -u iteblog hadoop fs -cat /user/iteblog/important/important-file.txt
important data
```

通过上面几步，我们已经恢复了误删除的重要文件。

通过编辑日志恢复

关于 Hadoop

的编辑日志介绍请参见：[《Hadoop文件系统元数据fsimage和编辑日志edits》](#)。如果你的 Hadoop 集群没有开启回收站功能，也没有对重要的数据创建快照，这时候如果有人将一份非常重要的数据误删除

了，那我们如何恢复这些数据？答案

是通过修改编辑日志，

但是通过这种方法不一定能恢复已经被删除的文件，或者只能恢复一部分被删除的文件，也可能恢复全部误删除的数据，这个和

你的集群繁忙状态有很大的关系。而且通过这种方式

恢复误删除的文件代价很高，风险很大，

需要谨慎使用。下面我来介绍通过这种恢复删除数据的步骤。

删除文件

```
sudo -u iteblog hadoop fs -rmr -skipTrash /user/iteblog/important-file.txt
```

由于上面删除操作使用了 `-skipTrash`

参数，这意味着这个文件会被直接删除，并不会先放到回收站。

恢复数据

NameNode 在收到删除命令时，会先将这个命令写到编辑日志中，然后会告诉 DataNode 执行真正的文件删除操作。所以我们需要做的是立刻停止 NameNode 和 DataNode 节点，阻止删除命令的执行。然后找到执行 `rmr` 操作发生时间对应的编辑日志，假设是 `edits_inprogress_0000000000000001512`，这个文件是二进制的形式，我们需要通过 HDFS 自带的命令将这个文件转换成可读的形式，如下：

```
$ hdfs oev -i edits_inprogress_0000000000000001512 -o edits_inprogress_000000000000001512.xml
```

上面执行的结果是二进制的编辑日志被转换成我们人类可读的xml格式的文件，我们找到执行删除 `important-file.txt` 文件的命令记录：

```
<RECORD>
  <OPCODE>OP_DELETE</OPCODE>
  <DATA>
    <TXID>1624</TXID>
    <LENGTH>0</LENGTH>
    <PATH>/user/iteblog/important-file.txt</PATH>
```

```
<TIMESTAMP>1515724198362</TIMESTAMP>  
<RPC_CLIENTID>34809cac-a89f-4113-98b5-10c54d7aac1a</RPC_CLIENTID>  
<RPC_CALLID>1</RPC_CALLID>  
</DATA>  
</RECORD>
```

OP_DELETE 这个标记就是删除操作，我们将这个标记修改成比较安全的操作（比如OP_SET_PERMISSIONS），如果这个命令是在最后，可以直接删除，然后保存。再将修改后的编辑日志转换成计算机能够识别的格式：

```
$ hdfs oev -i edits_inprogress_0000000000000001512.xml -o edits_inprogress_000000000000  
001512 -p binary
```

最后启动 NameNode 和 DataNode 节点，后面就看你的造化了。

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: [【】](#)（）