# 副本写入之避免小文件

如果一个storage name设置为多副本，则一般会由某一个特定的server来创建一个主写副本，其他的server都为该文件的同步副本。这是为了避免一个文件被多个线程或server来写入。

首先我们得知道文件的写入规则：

server1

aaa\_1\_2

server2

aaa\_1\_2

client

因为aaa\_1\_2是由server1来创建的，所以aaa\_1\_2在server1中为主写副本，aaa\_1\_2在server2中为同步文件。

Fid中的时间字段：此处时间字段对于客户来说，只为查找路径，跨小时，没必要一定生成新文件。

假设，server1或server2在写入文件时出现重启（人为重启或挂机重启）的现象，在server1或server2上的aaa\_1\_2是否可以避免小文件产生呢？

**提前关闭文件的文件复用：**

**考虑server2挂掉：主写副本所在的server1正常。**

若此时关闭文件，生成新文件，那么新来的数据的副本数是可以得到保证的。因为新文件的副本可能生成在server3上。但是，在server1上的aaa\_1\_2就会提前关闭，导致可能出现小文件。

那么这个小文件还能继续复用吗？

答案是可以的！

我们可以在提前关闭server1中的aaa\_1\_2时，同时记录该文件的元数据信息。

若server2在副本恢复之前启动，等待server2做完文件一致性校验后，server1可以去重新复用这个aaa\_1\_2这个文件，server2可以继续作为同步副本来进行文件写入。

若server2在副本恢复之后启动，也就是说经过副本恢复。那么server1也同样可以复用aaa\_1\_2这个文件，只是副本所在的主机，需要路由规则来寻找。

**考虑server1挂掉：同步副本所在的server2正常**

此时关闭文件，和之前该类讨论的相似。

此处也存在小文件是否能复用的问题。

答案也是可以的！

同样也需要在关闭文件时，记录该文件的元数据。

若server1在副本恢复之前启动，等待server1做完文件一致性校验，则可以继续使用该文件进行服务。主写副本和同步副本所在的服务不变。

若server1在副本恢复之后启动，那么在复用该文件时，必须通过路由规则来计算出新的主写副本所在的server，该server可以优先使用aaa\_1\_2来进行服务。

建议1：此处不认为所有的小文件都应该去复用，应该有个基线，比如小于50%才能复用。

**不关闭文件的不完全写入：**

**考虑server2挂掉：主写副本所在的server1正常。**

若此时不关闭文件，则server1中的aaa\_1\_2是可以继续接收数据的，只是此时为不完全写入。虽然处于不完全写入，但是此时不会产生小文件。

若server2在副本恢复之前启动，那么server2只需要做文件一致性校验，就可以继续正常使用。

若server2在副本恢复之后启动，那么server2上的文件就不需要关心了。因为server1完成该文件的写入后会将aaa\_1\_2迁移到其他server中。

**考虑server1挂掉：同步副本所在的server2正常**

若此时不关闭文件，需要选举出一个新的主写副本来继续接收数据，此时为不完全写入。同样也不会产生小文件。

若server1在副本恢复之前启动，那么server1只需要做文件一致性校验，此处是否让server1继续成为主写副本，不太影响最终结果，只需要保证文件单线程写入即可。

若server1在副本恢复之后启动，那么server1上的文件就不需要关心了。因为server2完成该文件的写入后会将aaa\_1\_2迁移到其他的server中。

多副本同时挂掉和单副本一样。

需要在完成文件写入的时候，记录文件的元数据即可。重启服务的时候，可以继续接着上次的文件继续写入。

注意点：删除后的小文件不能复用，归并后的小文件不能复用。对于不完全写入时，副本恢复可以先跳过该类文件的迁移，等文件成功写完后，再进行恢复。