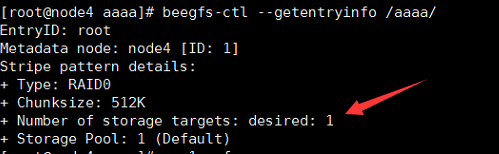
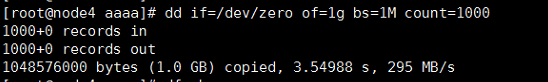
#### 一、 关于写一个文件会分成几份，落到那些storage上

1. 通过beegfs-ctl --getentryinfo /aaaa 可以看到挂载的池信息

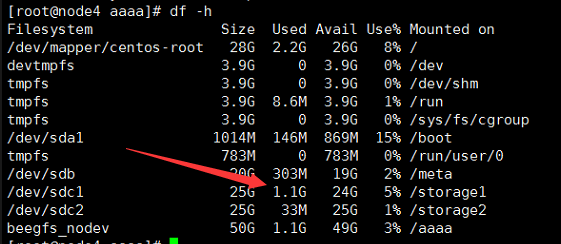
环境有2个storage，Numtarges为1表示不切片，即一个文件只会落到其中一个节点上



1. 这时写入一个1g文件



1. 用df可以看到写入了其中一个storage中，因为落点是哈希的，没有命令可以直接获取哈希后的落点，（如果熟悉元数据原理也可以推导出落点位置）



#### 二、关于冗余storage激活

1. 激活冗余storage

注释：

numtargets为冗余组的个数，由于我只有一组buddymirror，所以默认写1，如果有多个buddymirror组可以按具体数来赋值

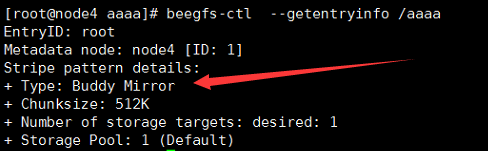
/aaaa为挂载目录

beegfs-ctl --setpattern --pattern=buddymirror --numtargets=1 /aaaa



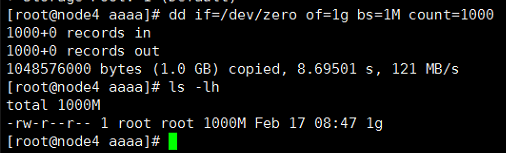
1. 查看当前挂载目录信息

beegfs-ctl --getentryinfo /aaaa

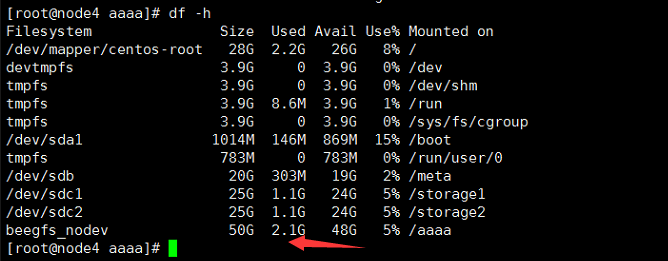


1. 验证

写一个1G文件



查看集群使用容量（已使用2G）

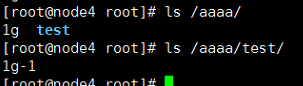


#### 三、关于文件定位（元数据原理）：

1. 首先根目录的ID固定为root，所以总是在38/51这个Hash目录中。无论是inodes还是dentry目录，里面都是有两级Hash目录，每一级有128个，根据DirEntry和DirInode的ID进行Hash，所以理论上可以做到很好的均衡。

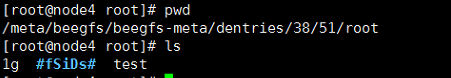


1. 我们在挂载目录下新建一个test目录，在其中新建一个1g-1文件



1. 当查找test/1g-1此文件时，会先依次寻找buddymir/inodes/38/51/root，

3.1 此目录中对于每个文件，有一个同名的文件，以及#fSiDs#子目录中的以ID为名称的硬链接，



3.2 对于子目录来说，有一个同名的文件，以及inodes目录中的以ID为名称的DirInode文件，ID是根据文件的OwnerID（即所在节点的NodeID）和EntryID（根据Hash确定目录路径）确定的，可以通过如下命令找到：

find /meta/beegfs/beegfs-meta/{inodes,dentries,buddymir} -type f



3.3 根据子目录id就可以在dentries中找到test目录的元数据(0-5E4A9DC1-1)，则此目录下的#fSiDs#目录中则是test目录下的文件id（0-5E4AA328-1），





1. 在storage节点中的对应文件就是具体的数据（我们配置的是冗余的storage池）



1. 如果有更深的目录则不断迭代，即可找到最终的文件。