

一、文件上传流程

- 1、client 获取要上传文件的 2 个 hashcode，传给 PS
- 2、ps 转发上传文件信息(文件名、文件大小、hashcode 等)给 NS
- 3、NS 根据文件 hashcode 检查待上传文件是否存在。
- 4、根据 hashcode 查找对应文件已上传成功的块，将 FileId 和上传成功的块信息返回给 PS。若未找到匹配的 hashcode，为待上传文件产生一个新的 FileId，并返回给 PS
- 5、PS 根据 NS 返回的已上传块信息，确定还需要续传哪些块，完成整个文件的上传，如没有要上传的块，跳到步骤 11
- 6、检查是否有需要上传的块，若有，PS 向 NS 发送创建块的命令
- 7、NS 根据 ds 的统计信息，优选 3 个（假设块副本数为 3）DS 返回给 PS，包括一台主 DS
- 8、ps 计算出该块在文件中的偏移，并通知 client 指定偏移的文件到 PS，PS 转发到主 DS
- 9、主 DS 完成块的获取后，返回块上传成功消息给 PS 和 NS
- 10、重复 6 ~ 10，当文件的所有块都上传成功，返回上传成功消息给 client

二、文件下载流程

- 1、client 发送下载请求到 PS，包括文件的 2 个 hashcode、下载起始偏移、下载长度
- 2、ps 转发 CLIENT 的下载请求到 NS
- 3、NS 检查 hashcode 是否存在；若不存在，直接返回下载失败。若存在，根据下载偏移和下载长度和文件总大小，即可确定要下载的文件在哪些块中，并将这些块和块内偏移返回给 ps。对于每个块，返回所有包含改块副本的 DS
- 5、检查是否有需要下载的块，若没有，跳到 10
(6 ~ 8 是块下载步骤)
- 6、ps 根据位置信息，选择一个 DS，发送 read chunk 请求，包括 FileID、ChunkId 和块内偏移
- 7、DS 从指定块的偏移处发送数据给 ps
- 8、ps 转发数据给 client
- 9、重复 6 ~ 7
- 10、返回下载完成

关于文件 hash,考虑文件 hash 的碰撞冲突问题，故为每个文件生成 1 个弱 hash 和 1 个强 hash(MD5)，弱 hash 的速度较快，但可能存在较高概率的碰撞。当校验文件时，先校验弱 HASH，若不能匹配，则证明文件不存在；若匹配，再检查强 hash。只有 2 个 hash 都同时匹配时，才表明该文件文件存在。