AlphaFileSystem 开发文档

1. 概述

文件系统能够实现创建文件，读文件，写文件，复制文件，查看文件数据块内容。

本文件系统使用了两个file-manager及3个block-manager，用户在进行操作时需要指定file-manager，不同的file-manager之间无法拥有读写对方文件的权限。

索引使用file-manager id和文件名结合进行。

文件的存储是分块存储的模式，默认块大小可通过DEFAULT\_BLOCK\_SIZE常量调整且可指定块大小。每个block所从属的block-manager通过随机数分配。block通过block-manager id及block id结合索引。block有根据block数据内容计算的校验码来在读取数据时判断block中的内容是否损坏。

文件的meta数据中存储的内容有文件的大小，块大小及含有文件数据的block列表。block的meta数据中存储block大小及block的校验码。

文件的读写支持从指定位置进行，File接口中存储有代表位置的指针，可以从控制台读取用户输入并调整该指针来进行位置的调整。如果用户输入了大于文件size的指针来进行文件写，则中间的字节将用0x00填充。

文件可以进行size的调整，减小size则将减小部分的数据所对应的块从文件的meta数据中删除，增加size则用0x00填充增加的部分。

文件的复制通过meta文件的复制进行。

理论上，block始终存在一致性，所有的操作都不对已经有过写操作的block进行更改，而通过对文件的meta数据的操作来调整文件内容。

系统支持duplication，通过LogicBlock实现，每个LogicBlock中存储有三个内容相同的block副本。

系统支持buffer，即系统中存储有最近使用的块的链表，当读或写文件时可以从链表中获取该块而无需进行文件IO。

1. 实现

分块存储通过对文件数据的拆分进行。遍历数据的byte数组并以blocksize常量切分数据，分块写入磁盘，且写入时自动创建多个副本存储进同一个LogicBlock。

对于从某个Block中部开始的写（包括setsize时大小从某个block中部截断），先用当前游标与blocksize取余数来获取offset，然后读取该offset长度的数据，将当前block从LogicBlock列表中删除并且重新将读取的部分数据写入文件系统。

读文件时先反序列化meta文件中的内容来初始化LogicBlockList，之后读取每个LogicBlock并初始化LogicBlock中的physicalblock，只要physical block校验成功则读取该block中的数据并跳出循环，否则继续校验下一个块，如果所有块都校验失败则抛出异常。

用户读和写文件时都可以设置游标来设置读写位置。若用户读文件时输入的长度与当前游标之和大于文件大小则抛出异常。

checksum的实现参考了lab的文档中的方法，在创建physical block时创建checknum，读取时重新计算并与meta文件中存取的checksum进行对比来确定块是否被损坏。

meta数据的存储参考了lab文档中的示例数据并实现了对该数据的序列化及反序列化。

buffer通过简单的链表实现，使用某个块时将该块放到链表头部，当链表大小到达阈值时把链表尾部的块写入实际存储单元。用户调用close时将链表清空，将所有块进行写入。

1. ·打开文件且读数据的过程概述

首先从控制台读取文件管理器和文件名和读取位置，创建一个AFSFile对象file。创建过程中根据meta文件初始化File的size，blocksize，LogicBlockList。调用move方法移动cursor，然后调用file的read方法。该方法根据游标定义到文件LogicBlockList的当前LogicBlock，依次读取每个LogicBlock，调用LogicBlock的getBlockList方法获取其中存储的PhysicalBlock，同理通过meta文件初始化PhysicalBlock，并调用PhysicalBlock的read方法读取有效块的数据，将有效块的数据依次写入结果，最后返回给主函数。

·写文件/创建文件的过程概述

调用AlphaWrite工具时首先从控制台读取文件管理器和文件名，判断该文件是否存在。若存在则进行下一步，不存在则调用AFSFileManager的newFile方法创建新的文件，对meta文件进行初始化。下一步是创建AFSFile对象，同样根据meta文件来初始化文件数据，根据游标找到需要写的位置。如果要写写的块内已有数据则将数据先读出，与即将放入新块的byte数组进行拼接，且不对之前的块进行更改。之后遍历数组，以块大小分割数组并创建新的PhysicalBlock，更改LogicBlockList将新块放入。