

Lab1 分布式文件系统

1. 实验背景与目标

1.1 背景说明

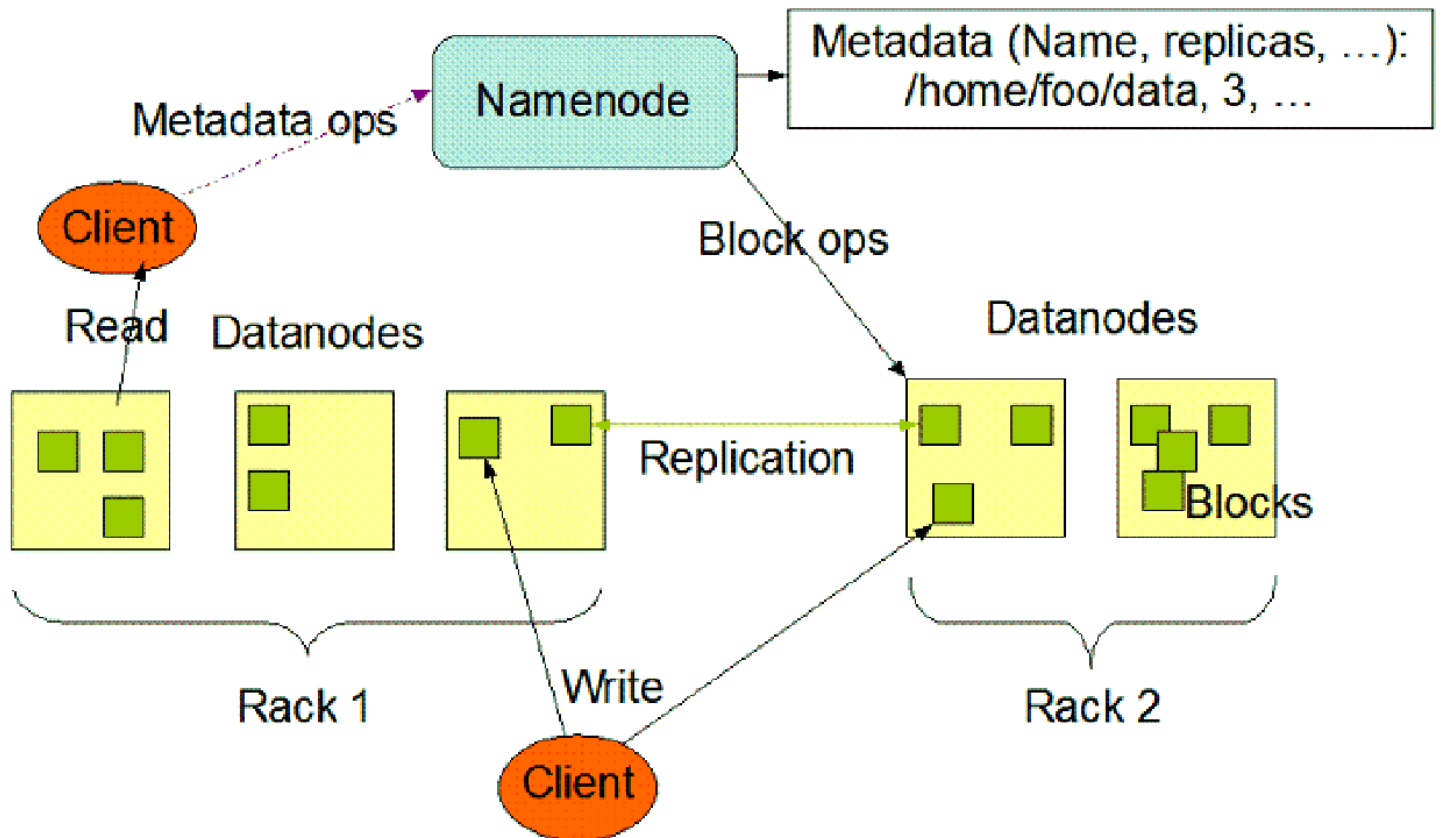
随着大数据应用场景日趋增多，分布式文件系统（如 Hadoop HDFS）在海量数据的存储与处理方面发挥了至关重要的作用。本次实验旨在通过一个简化的分布式文件系统实现，帮助同学们理解分布式存储架构的核心思想以及各模块之间的协作关系。实验中涉及 NameNode、DataNode 和 Client 模块的设计与实现，重点关注文件的分块存储以及持久化机制。

1.2 实验目标

实现一个基于 Java 的简化分布式文件系统，要求实现以下三个模块：

- **NameNode 板块**：负责管理文件元数据（文件名、总大小、数据块列表等），包括元数据持久化（写入 fsimage.txt 文件）与重启时数据恢复。
- **DataNode 板块**：负责存储文件数据，以固定大小（例如 4KB）的数据块保存数据，并实现分片写入：当单次写入数据超过当前块剩余空间时，新建数据块接续写入；同时，每个数据块按照多副本要求复制到多个 DataNode 上（实际实现中以不同文件名保存，如 replica0_文件名_blockX.txt、replica1_文件名_blockX.txt 等）。
- **Client 板块**：提供命令行交互接口，支持基本的文件操作（open、write、read、close），调用 NameNode 与 DataNode 提供的接口完成实际文件处理。

HDFS Architecture



2. 具体功能与评分细则

【功能要求】

1. 文件操作功能

◦ open:

- 若文件不存在且以写入模式（“w”）打开，则创建新文件并初始化第一个数据块；
- 若以读方式打开不存在的文件，则返回错误提示；
- 写入模式下，禁止多个进程同时写同一文件。

◦ write:

- 支持追加写入数据；
- 采用分片写入机制：在单次写入数据时，若当前数据块剩余空间不足，则先填满当前块，再自动新建块继续写入剩余数据；
- 每个数据块写入后按照多副本机制，生成多个副本文件，名称格式为 `replica{i}_文件名_block{块号}.txt`（副本数为3，例如）。

- **read:**
 - 能够顺序读取所有数据块内容，并拼接返回完整文件数据。
- **close:**
 - 关闭文件、更新写入状态和文件总大小；
 - 调用 NameNode 持久化元数据，将所有文件数据写入 fsimage.txt。

2. 持久化与多副本管理

- NameNode 在每次 close 时将文件元数据写入 fsimage.txt；
- 系统重启时，NameNode 自动读取 fsimage.txt，并根据其中记录的块的信息，恢复文件数据。
- DataNode 的写操作在每次写入时需要生成多副本文件（例如 3 个副本）。

【评分细则】（各部分满分可依据实际情况调整）

- 文件操作（open/read/write/close）正确实现，满足基本功能需求：40%
 - 数据块分片写入机制（按 BLOCK_SIZE 判断是否新建数据块，正确处理单次写入数据的切分）：15%
 - 持久化存储与重启恢复功能（fsimage.txt 与 data 目录数据保存、加载正确）：15%
 - 多副本管理（每个数据块至少生成设定数量（例如 3 个）副本，命名规则正确）：15%
 - 实验报告（需包含设计思路、实现以及功能演示）：15%
-

3. 实验提示

- **NameNode 部分**
 - 可以使用 HashMap 存储文件名到文件元数据的映射，文件元数据包含文件名、大小、数据块列表及当前写状态。
 - 持久化时，可以使用简单的文本格式，每行记录文件名、总大小和数据块数。
- **DataNode 部分**
 - 定义一个 BLOCK_SIZE 常量（例如：4 * 1024，单位模拟为字符数），在 write 方法中先判断当前数据块剩余空间。
 - 对于超过剩余空间的写入，请实现分片写入：利用循环先填满当前块，再新建块写入剩余部分。
 - 每次写入操作完成后，调用 persistBlock() 将该块的内容存储到对应磁盘文件中。
- **Client 部分**

- 提供命令行解析，将 open、write、read、close 命令映射到相应的模块方法；
- 输入错误时能给出提示信息，避免异常直接中断程序。

4. 示例Example（仅作参考）

注：实验功能满足评分要求即可，Client具体实现不需要按照以下格式

代码块

```
1  C:\Users\16004\Desktop\DSLlab1>java SimpleDistributedFS
2  无持久化元数据文件 fsimage.txt，无需加载。
3  欢迎使用支持持久化和多副本的简易分布式文件系统，请输入命令
   (open/read/write/close/exit):
4  > open test.txt
5  命令格式: open <filename> <mode>
6
7  > open test.txt w
8  文件 test.txt 已创建。
9  文件 test.txt 已以模式 w 打开。
10
11 > write MyGO!!!!
12 块 0 的数据持久化到 data\replica0_test.txt_block0.txt
13 块 0 的数据持久化到 data\replica1_test.txt_block0.txt
14 块 0 的数据持久化到 data\replica2_test.txt_block0.txt
15 当前块 0 大小 = 9
16 数据已追加到文件 test.txt
17
18 > close
19 文件 test.txt 已关闭，总大小: 9
20 FsImage 持久化保存成功到 fsimage.txt
21
22 > read test.txt
23 请先打开文件。
24
25 > open test.txt r
26 文件 test.txt 已以模式 r 打开。
27
28 > read test.txt
29 文件内容:
30 MyGO!!!!
31
32 > close
33 文件 test.txt 已关闭，总大小: 9
34 FsImage 持久化保存成功到 fsimage.txt
```

```
35
36 > exit
37 退出系统。
38
```

5. 实验提交

- 实验报告，介绍你的设计思路、实现以及功能演示。
- 完整实现代码