分布式系统概论 Lab4

指导老师：冯红伟

30/5/2023 23:59(GMT+8)

Contents

在本次实验中，你需要实现:

1. 5‘ 分布式协议Raft的 `AppendEntries` 功能，通过 *MIT 6.5840 Lab 2 Part 2B: log*[1]中的测试 `Test (2B): basic agreement` 。
2. 5’ 完善 Lab1 实现的 HDFS 应用，实现多副本功能和两阶段提交

1 Raft AppendEntries

1.1 实验配置

1.1.1 Setup go

本实验代码和测试使用Go语言[2]。

1.1.2 Pull code

```
$ git clone git://g.csail.mit.edu/6.5840-golabs-2023 6.5840
$ cd 6.5840
$ ls
Makefile src
$
```

1.1.3 Getting started

实现了`AppendEntries`后，运行`src/raft/test_test.go`中的测试用例，具体地，在命令行执行以下指令

```
$ cd ~/6.5840
$ git pull
...
```

[1] https://pdos.csail.mit.edu/6.824/labs/lab-raft.html
[2] 如果没有安装过，参考https://pdos.csail.mit.edu/6.824/labs/go.html

```
$ cd src/raft
$ go test -run 2B
Test (2B): basic agreement ...
  ... Passed --   0.9  3   16    4572    3
$
```

1.2 测试评分

```
$ go test -run 2B
Test (2B): basic agreement ...
  ... Passed --   0.9  3   16    4572    3
```

通过 `basic agreement` 测试用例则获得 5'

2 HDFS 多副本

2.1 需求

在Lab1实现的简单HDFS系统基础上，实现以下功能:

1. 增加 replication 副本功能，rep = 4 。即每个 block 在 HDFS 中有 4 个副本，分布在不同的 DataNode 上。每个 DataNode 至多保存一个文件的一个副本，即一个文件的两个相同副本不会出现在同一个 DataNode 上。

2. 副本更新的两阶段提交。即当一个block被修改时，需要先向所有副本发送 prepare 消息，等待所有副本回复 ok 后，再向所有副本发送 commit 消息，完成更新。如果有任何一个副本回复 abort 或者超时，那么向所有副本发送 abort 消息，取消更新。

3. 为方便测试，为 DataNode 实现 `dump()` 功能，即在收到请求并完成响应后向命令行打印当前存储的所有块信息。

注意事项:

- 在代码中适当添加注释，说明你的实现思路和细节
- 报告中描述你的设计方案，包括如何选择副本节点，如何处理故障恢复，如何保证一致性等 [3]
- 报告中简单说明你的 HDFS 应用在命令行界面如何操作，包括启动、读写和退出命令

2.2 测试用例

在命令行执行以下命令，启动应用

```
# terminal 1
cd src && mkdir bin
idlj -fall api.idl
javac src/*.java src/api/*.java src/impl/*.java src/utils/*java -d bin/
orbd -ORBInitialPort 1050 -ORBInitialHost localhost

#terminal 2 在目录 NameNode 下，启动NameNode
java -cp bin/ NameNodeLancher -ORBInitialPort 1050 -ORBInitialHost localhost

#terminal 3 在目录 DataNode1 下，启动DataNode1
java -cp bin/ DataNodeLancher -ORBInitialPort 1050 -ORBInitialHost localhost

#terminal 4 在目录 DataNode2 下，启动DataNode2
java -cp bin/ DataNodeLancher -ORBInitialPort 1050 -ORBInitialHost localhost

#terminal 5 在目录 DataNode3 下，启动DataNode3
```

[3] Lab4 的 Raft 部分不需要编写实验报告

```
java -cp bin/ DataNodeLancher -ORBInitialPort 1050 -ORBInitialHost localhost

#terminal 6 在目录 DataNode4 下，启动DataNode4
java -cp bin/ DataNodeLancher -ORBInitialPort 1050 -ORBInitialHost localhost

#terminal 7 在目录 Client1 下，启动Client1
java -cp bin/ ClientLancher -ORBInitialPort 1050 -ORBInitialHost localhost

#terminal 8 在目录 Client2 下，启动Client2
java -cp bin/ ClientLancher -ORBInitialPort 1050 -ORBInitialHost localhost
```

在命令行启动客户端后，在 Client1 执行以下操作

```
>> open test.txt w
INFO: fd=1
>> append 1 hello world
INFO: write done
>> close 1
INFO: fd 1 closed
>> open test.txt rw
INFO: fd=2
>> read 2
hello world
# 观察 DataNode 输出，杀死包含了 text.txt 文件 block 的 DataNode 只
剩下一个副本
# 再执行 read（1’）
>> read 2
hello world
# 由于不能获得所有副本的 ok ，写入失败（2’）
```

```
>> append 2 +++ hello world
INFO: write failed
# 恢复刚刚杀死的 DataNode ，执行写操作
>> append 2 +++ hello world
INFO: write done
# 同时在 Client2 执行读操作，两个 Client 都打印同样的内容
>> read 2
hello world
+++ hello world
# 同时在 Client2 执行写操作
>> append 2 ++++ hello world
# 两个 Client 有且仅有一个成功（2’）
INFO: write done/INFO: write failed
>> exit
INFO: bye
```

操作期间将检查 DataNode 的 `dump()` 输出结果，记分点已在测试用例中给出

3 提交

将 Raft 项目文件夹`6.5840`和 HDFS 项目文件夹一起打包成`<学号><姓名>Lab4.zip`文件 [4] 后，上传elearning

3.1 截止日期

30/5/2023 23:59(GMT+8)

[4] 例如：19302010001王明Lab4.zip