

## **PUSH BASED**

UPDATING FILE REPOSITORY VERSION to 0

2022-03-30 23:47:14,529 [INFO|project3.pa1.RMIServerInterfaceImpl|SuperPeerInterfaceImpl]

[INVALIDATION-QUERY] Received Query message from localhost:1234 InvalidateMessage

[key=cc72ff15-4b19-4f15-b71b-aaac2b0f9658.txt versionNumber -1019] at localhost port 1234

2022-03-30 23:47:14,543

[INFO|project3.pa1.RMIServerInterfaceImpl|LeafNodeServerInterfaceImpl] [FileRepository

(localhost,4542) Success Cache cleared] request from localhost:1234InvalidateMessage

[key=cc72ff15-4b19-4f15-b71b-aaac2b0f9658.txt versionNumber -1019]

2022-03-30 23:47:14,550

[INFO|project3.pa1.RMIServerInterfaceImpl|LeafNodeServerInterfaceImpl] [FileRepository

(localhost,5334) Success Cache cleared] request from localhost:1234InvalidateMessage

[key=cc72ff15-4b19-4f15-b71b-aaac2b0f9658.txt versionNumber -1019]

2022-03-30 23:47:14,572

[INFO|project3.pa1.RMIServerInterfaceImpl|LeafNodeServerInterfaceImpl] [FileRepository

(localhost,4545) Success Cache cleared] request from localhost:1234InvalidateMessage

[key=cc72ff15-4b19-4f15-b71b-aaac2b0f9658.txt versionNumber -1019]

.Initializing testing environment

setup

/mnt/c/Users/prash/eclipse-java-workspace/CS505-ASST3/deliver/project3/Gnutella-1/RMIServe  
r1/files

cc72ff15-4b19-4f15-b71b-aaac2b0f9658.txt

Added NEW TEST FILES ['cc72ff15-4b19-4f15-b71b-aaac2b0f9658.txt']

1. [Executed RETRIEVE AS Gnutella 1, Server 4 output: PUSH\_BASED\_CONSISTENCY  
TRUE

GNUTELLA CLUSTER 1 RMIServerChoice4

2022-03-30 23:47:09,630 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #4

RUNNING RETRIEVE :FileRepositoryFile cc72ff15-4b19-4f15-b71b-aaac2b0f9658.txt

Downloaded at localhost:5334

2022-03-30 23:47:09,630 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #4

RUNNING RETRIEVE :FileRepositoryFile cc72ff15-4b19-4f15-b71b-aaac2b0f9658.txt

Downloaded at localhost:5334

]

2. [Executed RETRIEVE AS Gnutella 1, Server 3 output: PUSH\_BASED\_CONSISTENCY  
TRUE

GNUTELLA CLUSTER 1 RMIServerChoice3

2022-03-30 23:47:10,397 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #3

RUNNING RETRIEVE :FileRepositoryFile cc72ff15-4b19-4f15-b71b-aaac2b0f9658.txt

Downloaded at localhost:4542

2022-03-30 23:47:10,397 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #3

RUNNING RETRIEVE :FileRepositoryFile cc72ff15-4b19-4f15-b71b-aaac2b0f9658.txt

Downloaded at localhost:4542

]

3. [Executed SEARCH AS Gnutella 1, Server 2 output: PUSH\_BASED\_CONSISTENCY TRUE

GNUTELLA CLUSTER 1 RMIServerChoice2  
 2022-03-30 23:47:10,661 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #2  
 RUNNING SEARCH :localhost:2345  
 2022-03-30 23:47:10,661 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #2  
 RUNNING SEARCH :localhost:2345  
 ]  
 5. [Executed REGISTER AS Gnutella 1, Server 1 output: PUSH\_BASED\_CONSISTENCY  
 TRUE  
 GNUTELLA CLUSTER 1 RMIServerChoice1  
 2022-03-30 23:47:10,924 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #1  
 RUNNING REGISTRY :ACCEPT  
 2022-03-30 23:47:10,924 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #1  
 RUNNING REGISTRY :ACCEPT  
 for file cc72ff15-4b19-4f15-b71b-aaac2b0f9658.txt ]  
 6. [Executed SEARCH AS Gnutella 1, Server 2 output: PUSH\_BASED\_CONSISTENCY TRUE  
 GNUTELLA CLUSTER 1 RMIServerChoice2  
 2022-03-30 23:47:11,207 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #2  
 RUNNING SEARCH :localhost:2345  
 2022-03-30 23:47:11,207 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #2  
 RUNNING SEARCH :localhost:2345  
 ]  
 7. [Executed RETRIEVE AS Gnutella 1, Server 3 output: PUSH\_BASED\_CONSISTENCY  
 TRUE  
 GNUTELLA CLUSTER 1 RMIServerChoice3  
 2022-03-30 23:47:12,007 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #3  
 RUNNING RETRIEVE :File Already Latest  
 2022-03-30 23:47:12,007 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #3  
 RUNNING RETRIEVE :File Already Latest  
 ]  
 10. [Executed SEARCH AS Gnutella 2, Server 3 output: PUSH\_BASED\_CONSISTENCY  
 TRUE  
 GNUTELLA CLUSTER 2 RMIServerChoice3  
 2022-03-30 23:47:12,849 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #3  
 RUNNING RETRIEVE :FileRepositoryFile cc72ff15-4b19-4f15-b71b-aaac2b0f9658.txt  
 Downloaded at localhost:4545  
 2022-03-30 23:47:12,849 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #3  
 RUNNING RETRIEVE :FileRepositoryFile cc72ff15-4b19-4f15-b71b-aaac2b0f9658.txt  
 Downloaded at localhost:4545  
 ]  
 11. [EXECUTED REGISTRY AS Gnutella 1, Server 1 output: PUSH\_BASED\_CONSISTENCY  
 TRUE  
 GNUTELLA CLUSTER 1 RMIServerChoice1  
 2022-03-30 23:47:13,159 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #1  
 RUNNING DEREGISTER :ACCEPT

2022-03-30 23:47:13,159 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #1  
RUNNING DEREGISTER :ACCEPT

]

12. [EXECUTED DEREGISTRY AS Gnutella 1, Server 2 output:

PUSH\_BASED\_CONSISTENCY TRUE

GNUTELLA CLUSTER 1 RMIServerChoice2

2022-03-30 23:47:13,455 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #2

RUNNING DEREGISTER :ACCEPT

2022-03-30 23:47:13,455 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #2

RUNNING DEREGISTER :ACCEPT

]

13. [EXECUTED DEREGISTRY AS Gnutella 1, Server 3 output:

PUSH\_BASED\_CONSISTENCY TRUE

GNUTELLA CLUSTER 1 RMIServerChoice3

2022-03-30 23:47:13,730 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #3

RUNNING DEREGISTER :ACCEPT

2022-03-30 23:47:13,730 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #3

RUNNING DEREGISTER :ACCEPT

]

14. [EXECUTED DEREGISTRY AS Gnutella 1, Server 4 output:

PUSH\_BASED\_CONSISTENCY TRUE

GNUTELLA CLUSTER 1 RMIServerChoice4

2022-03-30 23:47:13,986 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #4

RUNNING DEREGISTER :ACCEPT

2022-03-30 23:47:13,986 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #4

RUNNING DEREGISTER :ACCEPT

]

15. [EXECUTED DEREGISTRY AS Gnutella 1, Server 1 output:

PUSH\_BASED\_CONSISTENCY TRUE

GNUTELLA CLUSTER 1 RMIServerChoice1

2022-03-30 23:47:14,231 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #1

RUNNING DEREGISTER :ACCEPT

2022-03-30 23:47:14,231 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #1

RUNNING DEREGISTER :ACCEPT

]

FILE OUT OF DATE

**1. [Executed EDIT MESSAGE cc72ff15-4b19-4f15-b71b-aaac2b0f9658.txt On**

**GNUTELLA-1, RMIServer1 output: PUSH\_BASED\_CONSISTENCY TRUE**

**GNUTELLA CLUSTER 1 RMIServerChoice1**

**2022-03-30 23:47:14,519 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #1**

**RUNNING EDIT :ACK**

**2022-03-30 23:47:14,519 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #1**

**RUNNING EDIT :ACK**

**]**

## comments

The important lines are highlighted. Here a edit message is called and as we can see from the underlined lines, that all the files that could contain this file from the retrieve operation does in Push\_based\_integration\_test.py are being invalidated.

## PULL BASED APPROACH

2-03-30 23:51:07,953 [INFO|project3.pa1.RMIServerInterfaceImpl|RMIServerInterfaceImpl]

GET: key peer\_to\_contact localhost:2345

PSB FALSE

Master peer to contact localhost:2345

ready to download downloads/d8ff7f48-01af-46e6-93c3-f266e5c8858f.txt

This filename files/d8ff7f48-01af-46e6-93c3-f266e5c8858f.txt

KEYS being monitored [d8ff7f48-01af-46e6-93c3-f266e5c8858f.txt]

2022-03-30 23:51:07,969

[INFO|project3.pa1.RMIServerInterfaceImpl|CompletableFuture\$UniAccept] FileRepositoryFile  
**d8ff7f48-01af-46e6-93c3-f266e5c8858f.txt Downloaded at localhost:5334**

**POLLING RESULTS VALID**

**POLLING RESULTS VALID**

**POLLING RESULTS VALID**

**POLLING RESULTS VALID**

**POLLING RESULTS VALID**

**POLLING RESULTS VALID**

**POLLING RESULTS VALID**

.Initializing testing environment

setup

/mnt/c/Users/prash/eclipse-java-workspace/CS505-ASST3/deliver/project3/Gnutella-1/RMIServe  
r1/files

d8ff7f48-01af-46e6-93c3-f266e5c8858f.txt

Added NEW TEST FILES ['d8ff7f48-01af-46e6-93c3-f266e5c8858f.txt']

1. [Executed RETRIEVE AS Gnutella 1, Server 4 output: PUSH\_BASED\_CONSISTENCY  
FALSE

GNUTELLA CLUSTER 1 RMIServerChoice4

2022-03-30 23:50:52,723 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #4

RUNNING RETRIEVE :FileRepositoryFile d8ff7f48-01af-46e6-93c3-f266e5c8858f.txt

Downloaded at localhost:5334

2022-03-30 23:50:52,723 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #4

RUNNING RETRIEVE :FileRepositoryFile d8ff7f48-01af-46e6-93c3-f266e5c8858f.txt

Downloaded at localhost:5334

]

2. [Executed RETRIEVE AS Gnutella 1, Server 3 output: PUSH\_BASED\_CONSISTENCY  
FALSE

GNUTELLA CLUSTER 1 RMIServerChoice3

2022-03-30 23:50:53,521 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #3  
 RUNNING RETRIEVE :FileRepositoryFile d8ff7f48-01af-46e6-93c3-f266e5c8858f.txt  
 Downloaded at localhost:4542

2022-03-30 23:50:53,521 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #3  
 RUNNING RETRIEVE :FileRepositoryFile d8ff7f48-01af-46e6-93c3-f266e5c8858f.txt  
 Downloaded at localhost:4542

]

3. [Executed SEARCH AS Gnutella 1, Server 2 output: PUSH\_BASED\_CONSISTENCY FALSE  
 GNUTELLA CLUSTER 1 RMIServerChoice2  
 2022-03-30 23:50:53,770 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #2  
 RUNNING SEARCH :localhost:2345  
 2022-03-30 23:50:53,770 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #2  
 RUNNING SEARCH :localhost:2345

]

5. [Executed REGISTER AS Gnutella 1, Server 1 output: PUSH\_BASED\_CONSISTENCY  
 FALSE  
 GNUTELLA CLUSTER 1 RMIServerChoice1  
 2022-03-30 23:50:54,042 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #1  
 RUNNING REGISTRY :ACCEPT  
 2022-03-30 23:50:54,042 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #1  
 RUNNING REGISTRY :ACCEPT  
 for file d8ff7f48-01af-46e6-93c3-f266e5c8858f.txt ]

6. [Executed SEARCH AS Gnutella 1, Server 2 output: PUSH\_BASED\_CONSISTENCY FALSE  
 GNUTELLA CLUSTER 1 RMIServerChoice2  
 2022-03-30 23:50:54,290 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #2  
 RUNNING SEARCH :localhost:2345  
 2022-03-30 23:50:54,290 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #2  
 RUNNING SEARCH :localhost:2345

]

7. [Executed RETRIEVE AS Gnutella 1, Server 3 output: PUSH\_BASED\_CONSISTENCY  
 FALSE  
 GNUTELLA CLUSTER 1 RMIServerChoice3  
 2022-03-30 23:50:55,057 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #3  
 RUNNING RETRIEVE :File Already Latest  
 2022-03-30 23:50:55,057 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #3  
 RUNNING RETRIEVE :File Already Latest

]

10. [Executed RETRIEVE AS Gnutella 2, Server 3 output: PUSH\_BASED\_CONSISTENCY  
 FALSE  
 GNUTELLA CLUSTER 2 RMIServerChoice3  
 2022-03-30 23:50:55,825 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #3  
 RUNNING RETRIEVE :FileRepositoryFile d8ff7f48-01af-46e6-93c3-f266e5c8858f.txt  
 Downloaded at localhost:4545

2022-03-30 23:50:55,825 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #3  
RUNNING RETRIEVE :FileRepositoryFile d8ff7f48-01af-46e6-93c3-f266e5c8858f.txt  
Downloaded at localhost:4545

]

11. [EXECUTED REGISTRY AS Gnutella 1, Server 1 output: PUSH\_BASED\_CONSISTENCY  
FALSE

GNUTELLA CLUSTER 1 RMIServerChoice1

2022-03-30 23:50:56,070 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #1  
RUNNING DEREGISTER :ACCEPT

2022-03-30 23:50:56,070 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #1  
RUNNING DEREGISTER :ACCEPT

]

12. [EXECUTED DEREGISTRY AS Gnutella 1, Server 2 output:  
PUSH\_BASED\_CONSISTENCY FALSE

GNUTELLA CLUSTER 1 RMIServerChoice2

2022-03-30 23:50:56,346 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #2  
RUNNING DEREGISTER :ACCEPT

2022-03-30 23:50:56,346 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #2  
RUNNING DEREGISTER :ACCEPT

]

13. [EXECUTED DEREGISTRY AS Gnutella 1, Server 3 output:  
PUSH\_BASED\_CONSISTENCY FALSE

GNUTELLA CLUSTER 1 RMIServerChoice3

2022-03-30 23:50:56,611 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #3  
RUNNING DEREGISTER :ACCEPT

2022-03-30 23:50:56,611 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #3  
RUNNING DEREGISTER :ACCEPT

]

14. [EXECUTED DEREGISTRY AS Gnutella 1, Server 4 output:  
PUSH\_BASED\_CONSISTENCY FALSE

GNUTELLA CLUSTER 1 RMIServerChoice4

2022-03-30 23:50:56,879 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #4  
RUNNING DEREGISTER :ACCEPT

2022-03-30 23:50:56,879 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #4  
RUNNING DEREGISTER :ACCEPT

]

15. [EXECUTED DEREGISTRY AS Gnutella 1, Server 1 output:  
PUSH\_BASED\_CONSISTENCY FALSE

GNUTELLA CLUSTER 1 RMIServerChoice1

2022-03-30 23:50:57,115 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #1  
RUNNING DEREGISTER :ACCEPT

2022-03-30 23:50:57,115 [INFO|project3.pa1.RMIClient|RMIClient] Client on Server #1  
RUNNING DEREGISTER :ACCEPT

]

Sleep for 3 seconds to give enough time for the poller to catch up.  
FILE OUT OF DATE  
VALID

### **Comment**

The important lines are in bold which say that a poller is running on these recently downloaded files and if they differ from the master copy then they are invalidated. The same test was used for both push based approach and pull based approach.

### **Tests**

#### **Comment**

The test first retrieves 4 files on different peer nodes and then checks they are all invalidated due to an edit/delete being made on the master copy within the agreements of the software. These agreements include the TTR time which is important because if an update happens in between two TTR polls then the file will remain valid until the next TTR poll.

#### **File:**

```
#!/usr/bin/env python3
import os
import random
import sys
import unittest
import random
import uuid
import string
import filecmp
import subprocess
from base64 import decode
from profile import MyClass
import shutil
import json
```

```

class TestPeerToPeerMethods(unittest.TestCase):
    def setupClass(self):
        unittest.TestCase.setUp(self)
        self.__class__.myclass = MyClass()

    ClassIsSetup = False
    new_files = []
    def setUp(self):
        self.new_files = self.new_files
        if not self.ClassIsSetup:
            print("Initializing testing environment")
            # run the real setup
            self.new_files = []
            self.new_files_dir_appended = []
            length = 10000
            letters = string.ascii_lowercase
            print('setup')
            def get_parent_dir(directory):
                import os
                return os.path.dirname(directory)
            for dirs in os.walk((get_parent_dir(get_parent_dir(os.getcwd())) +
"/Gnutella-1/RMIServer1/files")):
                d = dirs[0]
                print(d)
                for i in range(1):
                    filename = str(uuid.uuid4()) + ".txt"
                    print(filename)
                    self.new_files.append(filename)
                    self.new_files_dir_appended.append(os.path.join(d, filename))
                    size = random.randint(1, 100)
                    with open(os.path.join(d, filename), "w") as f:
                        f.write("".join(random.choice(letters) for i in range(length)))
            print("Added NEW TEST FILES ", self.new_files)
            self.setupClass()
            # remember that it was setup already
            self.__class__.ClassIsSetup = True
            self.__class__.new_files = self.new_files

    def test_a_new_file_gets_added_single_gnutella_from_pa1(self):
        def get_parent_dir(directory):
            import os
            return os.path.dirname(directory)

```



```

env = {
    **os.environ,
    "PUSH_BASED_CONSISTENCY": "FALSE",
    "TTR": "450",
}

encoding = 'utf-8'
subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',
    '-DserverChoice=4',
    os.path.join(os.path.dirname(__file__),
        'RMISuperPeerClient.jar'), 'RETRIEVE',
    self.new_files[0]], stdout=subprocess.PIPE, env=env)
out, err = subprocess_output.communicate()
out = str(out.decode(encoding))
print('1. [Executed RETRIEVE AS Gnutella 1, Server 4 output:', out, ']')
#assert 'RETRIEVE :No such file on any peer' in out
self.new_files_dir_appended.append((get_parent_dir(get_parent_dir(os.getcwd())) +
    "/Gnutella-1/RMIServer4/download"
    + self.new_files[0]))
subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',
    '-DserverChoice=3',
    os.path.join(os.path.dirname(__file__),
        'RMISuperPeerClient.jar'), 'RETRIEVE',
    self.new_files[0]], stdout=subprocess.PIPE, env=env)
out, err = subprocess_output.communicate()
out = str(out.decode(encoding))
print('2. [Executed RETRIEVE AS Gnutella 1, Server 3 output:', out, ']')

self.new_files_dir_appended.append(os.path.join((get_parent_dir(get_parent_dir(os.getcwd()))
    + "/Gnutella-1/RMIServer3/download"
    ,self.new_files[0]))
#assert 'No such file on any peer' in out

subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',
    '-DserverChoice=2',
    os.path.join(os.path.dirname(__file__),
        'RMISuperPeerClient.jar'), 'SEARCH',
    self.new_files[0]], stdout=subprocess.PIPE, env=env)
out, err = subprocess_output.communicate()
out = str(out.decode(encoding))
print('3. [Executed SEARCH AS Gnutella 1, Server 2 output:', out, ']')
assert 'SEARCH :localhost:2345' in out

```

```

subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',
                                     '-DserverChoice=1',
                                     os.path.join(os.path.dirname(__file__),
                                     'RMISuperPeerClient.jar'), 'REGISTRY',
                                     self.new_files[0]], stdout=subprocess.PIPE, env=env)
out, err = subprocess_output.communicate()
out = str(out.decode(encoding))
print('5. [Executed REGISTER AS Gnutella 1, Server 1 output:', out, ' for file ',
self.new_files[0], ']')
assert 'ACCEPT' in out

subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',
                                     '-DserverChoice=2',
                                     os.path.join(os.path.dirname(__file__),
                                     'RMISuperPeerClient.jar'), 'SEARCH',
                                     self.new_files[0]], stdout=subprocess.PIPE, env=env)
out, err = subprocess_output.communicate()
out = str(out.decode(encoding))
print('6. [Executed SEARCH AS Gnutella 1, Server 2 output:', out, ']')
assert 'SEARCH :localhost:2345' in out

subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',
                                     '-DserverChoice=3',
                                     os.path.join(os.path.dirname(__file__),
                                     'RMISuperPeerClient.jar'), 'RETRIEVE',
                                     self.new_files[0]], stdout=subprocess.PIPE, env=env)
out, err = subprocess_output.communicate()
out = str(out.decode(encoding))

print('7. [Executed RETRIEVE AS Gnutella 1, Server 3 output:', out, ']')

assert 'File Already Latest' in out

self.new_files_dir_appended.append(os.path.join((get_parent_dir(get_parent_dir(os.getcwd()))
+ "/Gnutella-1/RMIServer3/download"
, self.new_files[0]))

subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=2',
                                     '-DserverChoice=3',
                                     os.path.join(os.path.dirname(__file__),
                                     'RMISuperPeerClient.jar'), 'RETRIEVE',
                                     self.new_files[0]], stdout=subprocess.PIPE, env=env)
out, err = subprocess_output.communicate()

```

```

out = str(out.decode(encoding))

print('10. [Executed RETRIEVE AS Gnutella 2, Server 3 output:', out, ']')
assert 'FileRepositoryFile '+ self.new_files[0]+' Downloaded at localhost:4545' in out

subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',
                                      '-DserverChoice=1',
                                      os.path.join(os.path.dirname(__file__),
                                                    'RMISuperPeerClient.jar'), 'DEREGISTER',
                                      self.new_files[0]], stdout=subprocess.PIPE, env=env)
out, err = subprocess_output.communicate()
out = str(out.decode(encoding))
print('11. [EXECUTED REGISTRY AS Gnutella 1, Server 1 output:', out, ']')
assert 'ACCEPT' in out


subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',
                                      '-DserverChoice=2',
                                      os.path.join(os.path.dirname(__file__),
                                                    'RMISuperPeerClient.jar'), 'DEREGISTER',
                                      self.new_files[0]], stdout=subprocess.PIPE, env=env)
out, err = subprocess_output.communicate()
out = str(out.decode(encoding))
print('12. [EXECUTED DEREGISTRY AS Gnutella 1, Server 2 output:', out, ']')
assert 'ACCEPT' in out


subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',
                                      '-DserverChoice=3',
                                      os.path.join(os.path.dirname(__file__),
                                                    'RMISuperPeerClient.jar'), 'DEREGISTER',
                                      self.new_files[0]], stdout=subprocess.PIPE, env=env)
out, err = subprocess_output.communicate()
out = str(out.decode(encoding))
print('13. [EXECUTED DEREGISTRY AS Gnutella 1, Server 3 output:', out, ']')
assert 'ACCEPT' in out


subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',
                                      '-DserverChoice=4',
                                      os.path.join(os.path.dirname(__file__),
                                                    'RMISuperPeerClient.jar'), 'DEREGISTER',
                                      self.new_files[0]], stdout=subprocess.PIPE, env=env)
out, err = subprocess_output.communicate()
out = str(out.decode(encoding))
print('14. [EXECUTED DEREGISTRY AS Gnutella 1, Server 4 output:', out, ']')

```

```
assert 'ACCEPT' in out
```

```
subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',  
                                     '-DserverChoice=1',  
                                     os.path.join(os.path.dirname(__file__),  
                                                  'RMISuperPeerClient.jar'), 'DEREGISTER',  
                                     self.new_files[0]], stdout=subprocess.PIPE, env=env)  
out, err = subprocess_output.communicate()  
out = str(out.decode(encoding))  
print('15. [EXECUTED DEREGISTRY AS Gnutella 1, Server 1 output:', out, ']')  
assert 'ACCEPT' in out
```

```
encoding = 'utf-8'
```

```
def get_parent_dir(directory):  
    import os  
    return os.path.dirname(directory)
```

```
encoding = 'utf-8'  
# Files are always added to the Gnutella server 1 , RMI Server 1  
# We go and check the file for existence  
# Check THE GNUTELLA-2 cluster files folder and it should be downloaded there  
# On executing the QUERY_MESSAGE command.  
# We also compare the file contents.
```

```
file_a_location = None  
for dirs in os.walk((get_parent_dir(get_parent_dir(os.getcwd())) +  
"/Gnutella-1/RMIServer1/file_metadata")):  
    d = dirs[0]  
    file_a_location = os.path.join(d, self.new_files[0] + ".json")
```

```
file_b_location = None  
for dirs in os.walk((get_parent_dir(get_parent_dir(os.getcwd())) +  
"/Gnutella-1/RMIServer3/file_metadata")):  
    d = dirs[0]  
    file_b_location = os.path.join(d, self.new_files[0] + ".json")
```

```
file_c_location = None  
for dirs in os.walk((get_parent_dir(get_parent_dir(os.getcwd())) +  
"/Gnutella-1/RMIServer4/file_metadata")):  
    d = dirs[0]  
    file_c_location = os.path.join(d, self.new_files[0] + ".json")
```

```
#####  
#POLL BASED  
#####
```

```
env = {  
    **os.environ,  
    "PUSH_BASED_CONSISTENCY": "FALSE",  
    "TTR": "450"  
}
```

```
origin_server_metadata = json.load(open(file_a_location))  
leaf_server_metadata_one = json.load(open(file_b_location))  
leaf_server_metadata_two = json.load(open(file_c_location))
```

```
assert origin_server_metadata['status'] == "VALID"  
assert origin_server_metadata['isMasterClient'] == True  
assert origin_server_metadata['version'] == 0  
assert leaf_server_metadata_one['status'] == "VALID"  
assert leaf_server_metadata_one['isMasterClient'] == False  
assert leaf_server_metadata_two['status'] == "VALID"  
assert leaf_server_metadata_one['isMasterClient'] == False
```

```
encoding = 'utf-8'  
subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',  
                                     '-DserverChoice=1',  
                                     os.path.join(os.path.dirname(__file__),  
                                                  'RMISuperPeerClient.jar'), 'EDIT',  
                                     self.new_files[0]], stdout=subprocess.PIPE, env=env)  
out, err = subprocess_output.communicate()  
out = str(out.decode(encoding))
```

```
from time import sleep  
print('Sleep for 3 seconds to give enough time for the poller to catch up.')  
sleep(3);
```

```
origin_server_metadata = json.load(open(file_a_location))  
leaf_server_metadata_one = json.load(open(file_b_location))  
leaf_server_metadata_two = json.load(open(file_c_location))
```

```
assert origin_server_metadata['status'] == "VALID"  
assert origin_server_metadata['isMasterClient'] == True
```

```

assert origin_server_metadata['version'] == 1
print(leaf_server_metadata_one['status'])
assert leaf_server_metadata_one['status'] == "FILE OUT OF DATE"
assert leaf_server_metadata_one['isMasterClient'] == False
assert leaf_server_metadata_two['status'] == "FILE OUT OF DATE"
assert leaf_server_metadata_one['isMasterClient'] == False


encoding = 'utf-8'
subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',
                                      '-DserverChoice=3',
                                      os.path.join(os.path.dirname(__file__),
                                                  'RMISuperPeerClient.jar'), 'REFRESH',
                                      self.new_files[0]], stdout=subprocess.PIPE, env=env)

out, err = subprocess_output.communicate()
out = str(out.decode(encoding))
origin_server_metadata = json.load(open(file_a_location))
leaf_server_metadata_one = json.load(open(file_b_location))
leaf_server_metadata_two = json.load(open(file_c_location))

assert origin_server_metadata['status'] == "VALID"
assert origin_server_metadata['isMasterClient'] == True
assert origin_server_metadata['version'] == 1
print(leaf_server_metadata_one['status'])
assert leaf_server_metadata_one['status'] == "VALID"
assert leaf_server_metadata_one['isMasterClient'] == False
assert leaf_server_metadata_two['status'] == "FILE OUT OF DATE"
assert leaf_server_metadata_one['isMasterClient'] == False


encoding = 'utf-8'
subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',
                                      '-DserverChoice=4',
                                      os.path.join(os.path.dirname(__file__),
                                                  'RMISuperPeerClient.jar'), 'REFRESH',
                                      self.new_files[0]], stdout=subprocess.PIPE, env=env)

out, err = subprocess_output.communicate()
out = str(out.decode(encoding))

from time import sleep
print('Sleep for 3 seconds to give enough time for the poller to catch up.')

```

```

sleep(3);

origin_server_metadata = json.load(open(file_a_location))
leaf_server_metadata_one = json.load(open(file_b_location))
leaf_server_metadata_two = json.load(open(file_c_location))

assert origin_server_metadata['status'] == "VALID"
assert origin_server_metadata['isMasterClient'] == True
assert origin_server_metadata['version'] == 1
print(leaf_server_metadata_one['status'])
assert leaf_server_metadata_one['status'] == "VALID"
assert leaf_server_metadata_one['isMasterClient'] == False
assert leaf_server_metadata_two['status'] == "VALID"
assert leaf_server_metadata_one['isMasterClient'] == False

print('1. [Executed EDIT MESSAGE', self.new_files[0], ' On GNUTELLA-1, RMIServer1
output:', out, ']')

def test_profile(self):
    env = {
        **os.environ,
        "PUSH_BASED_CONSISTENCY": "FALSE",
        "TTR": "450",
    }
    encoding = 'utf-8'
    subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',
                                          '-DserverChoice=1',
                                          os.path.join(os.path.dirname(__file__),
                                                        'RMISuperPeerClient.jar'), 'EDIT',
                                          self.new_files[0]], stdout=subprocess.PIPE, env=env)
    out, err = subprocess_output.communicate()
    out = str(out.decode(encoding))

    for i in range (30):
        encoding = 'utf-8'
        subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',
                                              '-DserverChoice=1',
                                              os.path.join(os.path.dirname(__file__),
                                                            'RMISuperPeerClient.jar'), 'EDIT',
                                              self.new_files[0]], stdout=subprocess.PIPE, env=env)
        out, err = subprocess_output.communicate()

```

```
out = str(out.decode(encoding))
subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',
                                      '-DserverChoice=4',
                                      os.path.join(os.path.dirname(__file__),
                                                    'RMISuperPeerClient.jar'), 'REFRESH',
                                      self.new_files[0]], stdout=subprocess.PIPE, env=env)
```

```
out, err = subprocess_output.communicate()
out = str(out.decode(encoding))
subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',
                                      '-DserverChoice=3',
                                      os.path.join(os.path.dirname(__file__),
                                                    'RMISuperPeerClient.jar'), 'REFRESH',
                                      self.new_files[0]], stdout=subprocess.PIPE, env=env)
```

```
subprocess_output = subprocess.Popen(['java', '-jar', '-DclientId=1',
                                      '-DserverChoice=4',
                                      os.path.join(os.path.dirname(__file__),
                                                    'RMISuperPeerClient.jar'), 'REFRESH',
                                      self.new_files[0]], stdout=subprocess.PIPE, env=env)
```

```
def suite():
    suite = unittest.TestSuite()
    if os.getenv("PROFILE") == "TRUE":
        suite.addTest(TestPeerToPeerMethods('test_profile'))
    else:
```

```
suite.addTest(TestPeerToPeerMethods('test_a_new_file_gets_added_single_gnutella_from_pa1
'))
```

```
    return suite
```

```
if __name__ == '__main__':
    runner = unittest.TextTestRunner()
    runner.run(suite())
```