Peer-to-Peer Systems and Overlay Networks

1º Semester – Academic Year 2013/2014

Kuganesan Srijeyanthan, 79531 - kuganesan.srijeyanthan@ist.utl.pt
Gayana Ranganatha Chandrasekara Pilana Withanage, 79529 - gayana.withanage@ist.utl.pt
Group 3 - Taguspark

Introduction

Peer to Peer distributed file system is an emerging method of utilizing commodity hardware with high degree of fault tolerance and scalability. Our group has researched, implemented some sample programs and tested with local virtual nodes in order to finalize the communication protocol for this application development. We have chosen Pastry [1, 2] as a main DHT service on top of virtual file system.

Protocol Selection and Design

Why Pastry

Refer the below table in order to compare some features of different DHT protocols and it will help to justify our selection of protocol PASTRY.

Protocol Features	Chord	Pastry	Kademlia
Routing	Log(n)	Log(n) but can be improved with neighbor and leaf nodes	Log(n)
Locality	No	Yes	No
Node Join Complexity	Log^{2}(n)	Log(n)	Log^{2}(n)
Open Source Implementation	Yes	Yes	Yes
File Systems related prior applications	No	Yes	No
Security Implementation	No	Yes	Yes

The following diagram clearly explains how file systems are going to be dispersed in to several nodes which are belongs to the Pastry ring. Let's assume UserA is going to send p2p_smaple.txt file via peer to peer network , once the file is created inside the fuse file system all the contents will be send via Pastry chunk by chunk, which we could be able set from configuration file.

DHT implementation of P2P distributed file system using Pastry

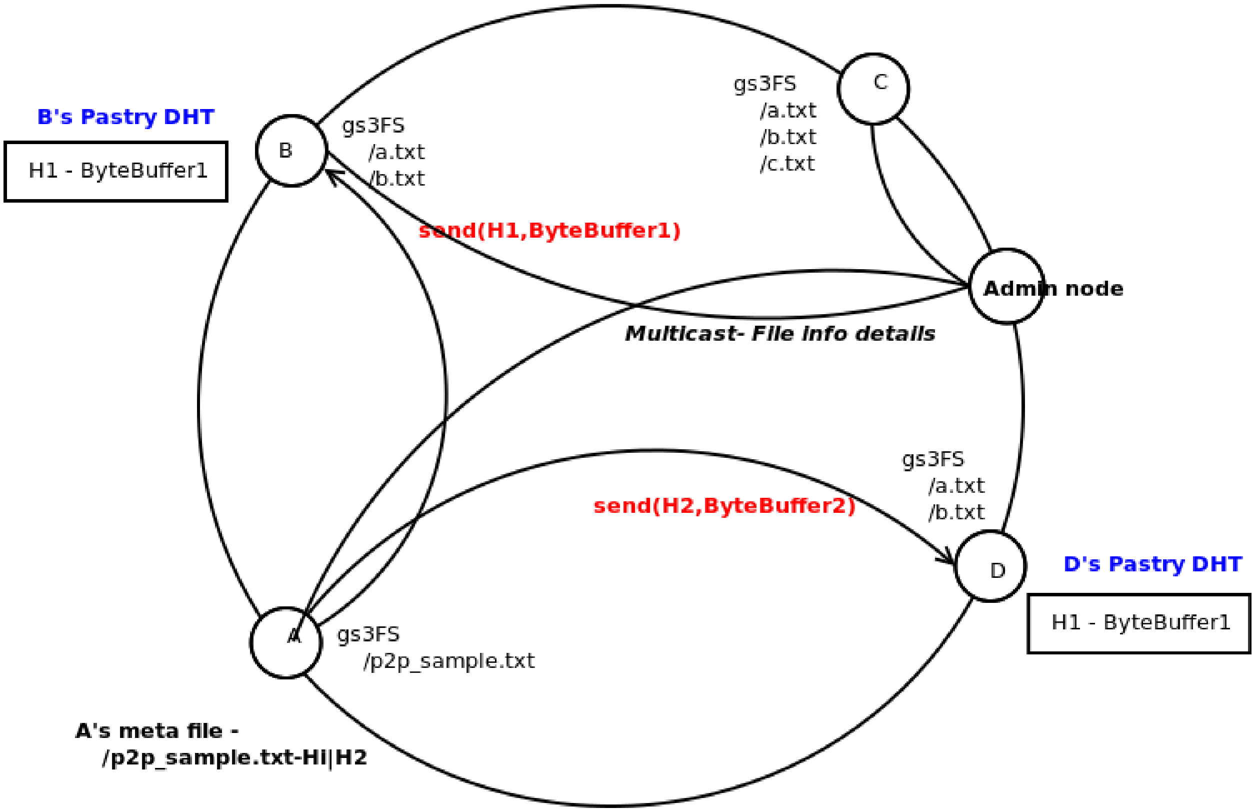


Fig.1 DHT architecture of P2P distributed file system

Architecture of the system

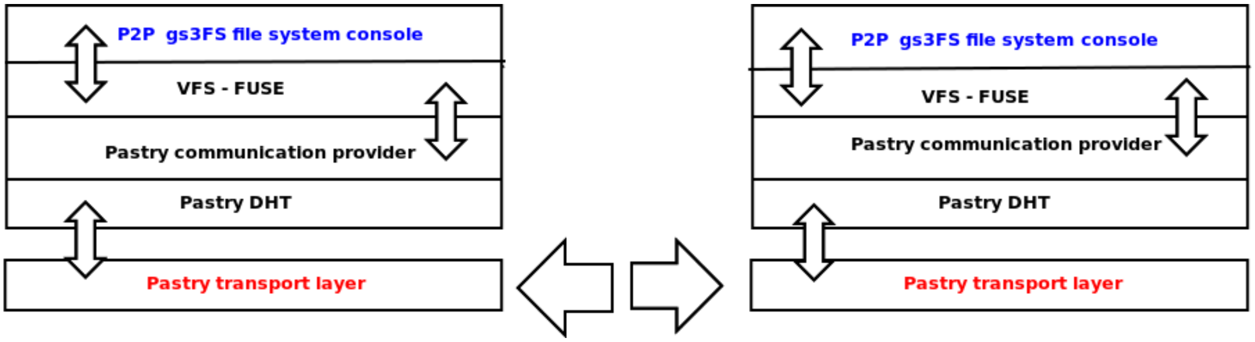

P2P gs3FS file system console
VFS - FUSE
Pastry communication provider
Pastry DHT
Pastry transport layer
P2P gs3FS file system console
VFS - FUSE
Pastry communication provider
Pastry DHT
Pastry transport layer

Fig.2 System architecture how message flows in between two nodes

Fig.2 shows the node architecture of P2P DFS implementation, all the operations are transparent to user and will be handled by Pastry communication provider module which we are going to implement. It will handle node entry, file management, and file replication according to Pastry protocol concepts.

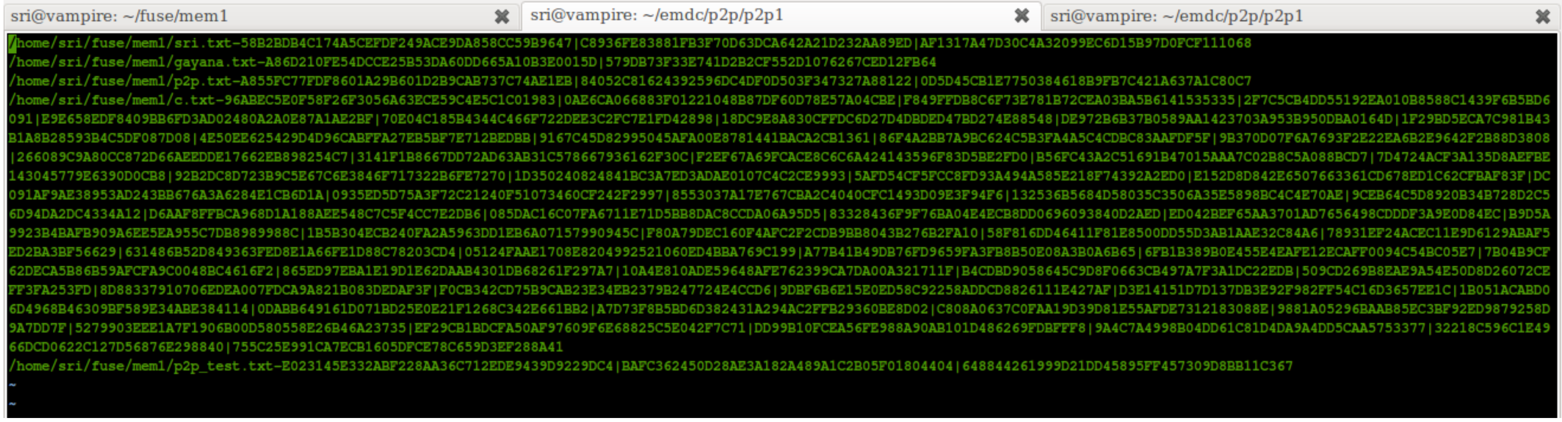
<u>File replication management</u>

File replication will be handled by the Pastry implementation according to the configuration value we provide. In our implementation we used the replication node count as 2 which will make 3 replicas where 2 in the leaf set and other in the node itself.

Metadata file

Each node has its own meta data file stored in to its persistent storage. Metafile will generate before the peer enter into the Pastry ring if the user is a new comer.
If a previous user joins again to the ring, application will check the availability of the Metafile and will request the files from peers and mount the files to the user's FUSE file system.
Metafile keep the records as <Key,Value> pairs where
Key = unique file path
Value = Pipe separated Hash value set of each file chunks of the desired size those user shared previously.

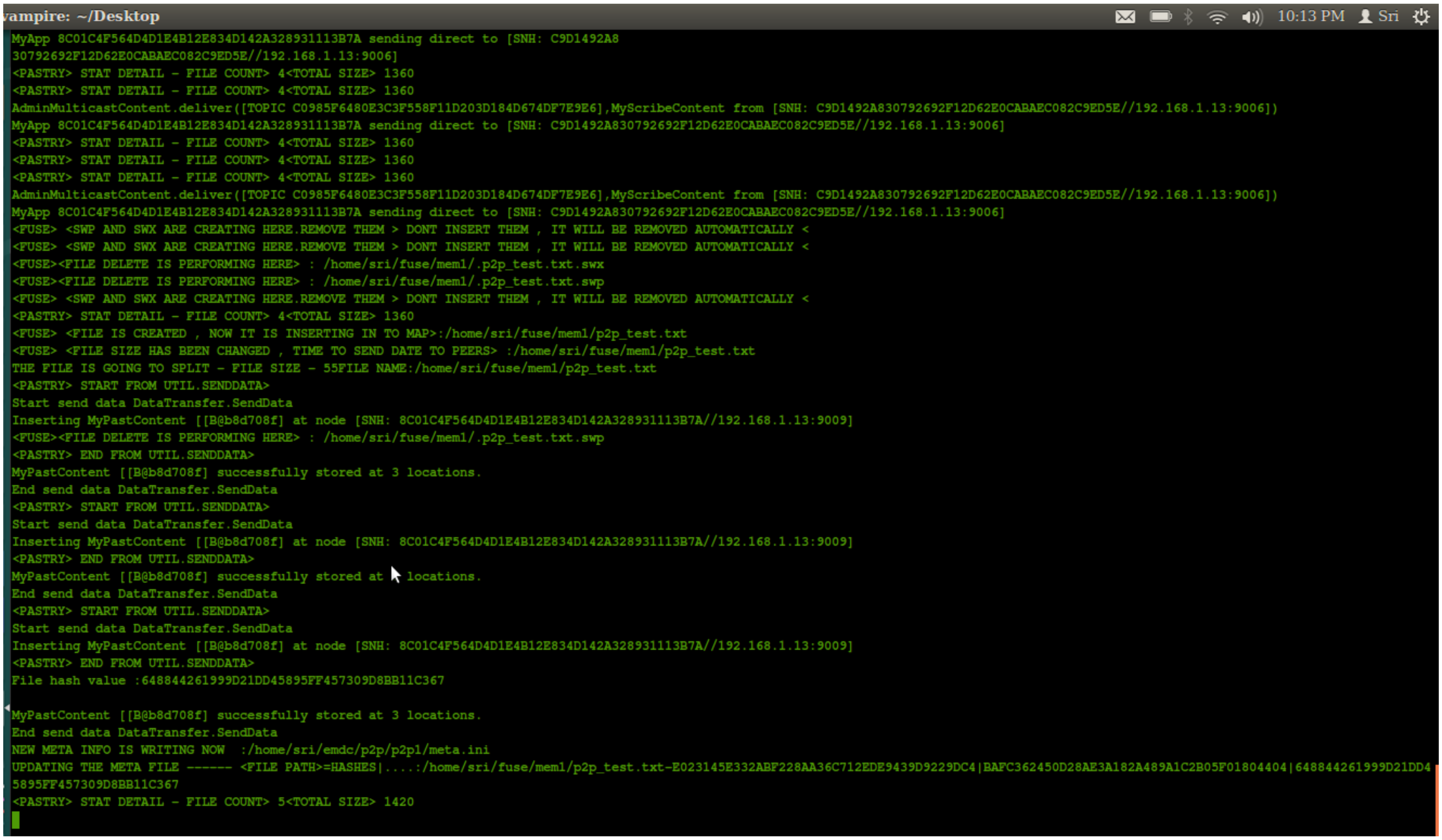
Below figure shows a sample Meta info file generated by our application.



```
/home/sri/fuse/mem1/sri.txt-58B2BDB4C174A5CEFDF249ACE9DA858CC59B9647|C8936FE83881FB3F70D63DCA642A21D232AA89ED|AF1317A47D30C4A32099EC6D15B97D0FCF111068
/home/sri/fuse/mem1/gayana.txt-A86D210FE54DCCE25B53DA60DD665A10B3E0015D|579DB73F33E741D2B2CF552D1076267CED12FB64
/home/sri/fuse/mem1/p2p.txt-A855FC77FDF8601A29B601D2B9CAB737C74AE1EB|84052C81624392596DC4DF0D503F347327A88122|0D5D45CB1E7750384618B9FB7C421A637A1C80C7
/home/sri/fuse/mem1/c.txt-96ABEC5E0F58F26F3056A63ECE59C4E5C1C01983|0AE6CA066883F01221048B87DF60D78E57A04CBE|F849FFDB8C6F73E781B72CEA03BA5B6141535335|2F7C5CB4DD55192EA010B8588C1439F6B5BD6
091|E9E658EDF8409BB6FD3AD02480A2A0E87A1AE2BF|70E04C185B4344C466F722DEE3C2FC7E1FD42898|18DC9E8A830CFFDC6D27D4DBDED47BD274E88548|DE972B6B37B0589AA1423703A953B950DBA0164D|1F29BD5ECA7C981B43
B1A8B28593B4C5DF087D08|4E50EE625429D4D96CABFFA27EB5BF7E712BEDBB|9167C45D82995045AFA00E8781441BACA2CB1361|86F4A2BB7A9BC624C5B3FA4A5C4CDBC83AAFDF5F|9B370D07F6A7693F2E22EA6B2E9642F2B88D3808
|266089C9A80CC872D66AEEDDE17662EB898254C7|3141F1B8667DD72AD63AB31C578667936162F30C|F2EF67A69FCACE8C6C6A424143596F83D5BE2FD0|B56FC43A2C51691B47015AAA7C02B8C5A088BCD7|7D4724ACF3A135D8AEFBE
143045779E6390D0CB8|92B2DC8D723B9C5E67C6E3846F717322B6FE7270|1D350240824841BC3A7ED3ADAE0107C4C2CE9993|5AFD54CF5FCC8FD93A494A585E218F74392A2ED0|E152D8D842E6507663361CD678ED1C62CFBAF83F|DC
091AF9AE38953AD243BB676A3A6284E1CB6D1A|0935ED5D75A3F72C21240F51073460CF242F2997|8553037A17E767CBA2C4040CFC1493D09E3F94F6|132536B5684D58035C3506A35E5898BC4C4E70AE|9CEB64C5D8920B34B728D2C5
6D94DA2DC4334A12|D6AAF8FFBCA968D1A188AEE548C7C5F4CC7E2DB6|085DAC16C07FA6711E71D5BB8DAC8CCDA06A95D5|83328436F9F76BA04E4ECB8DD0696093840D2AED|ED042BEF65AA3701AD7656498CDDDF3A9E0D84EC|B9D5A
9923B4BAFB909A6EE5EA955C7DB8989988C|1B5B304ECB240FA2A5963DD1EB6A07157990945C|F80A79DEC160F4AFC2F2CDB9BB8043B276B2FA10|58F816DD46411F81E8500DD55D3AB1AAE32C84A6|78931EF24ACEC11E9D6129ABAF5
ED2BA3BF56629|631486B52D849363FED8E1A66FE1D88C78203CD4|05124FAAE1708E8204992521060ED4BBA769C199|A77B41B49DB76FD9659FA3FB8B50E08A3B0A6B65|6FB1B389B0E455E4EAFE12ECAFF0094C54BC05E7|7B04B9CF
62DECA5B86B59AFCFA9C0048BC4616F2|865ED97EBA1E19D1E62DAAB4301DB68261F297A7|10A4E810ADE59648AFE762399CA7DA00A321711F|B4CDBD9058645C9D8F0663CB497A7F3A1DC22EDB|509CD269B8EAE9A54E50D8D26072CE
FF3FA253FD|8D88337910706EDEA007FDCA9A821B083DEDAF3F|F0CB342CD75B9CAB23E34EB2379B247724E4CCD6|9DBF6B6E15E0ED58C92258ADDCD8826111E427AF|D3E14151D7D137DB3E92F982FF54C16D3657EE1C|1B051ACABD0
6D4968B46309BF589E34ABE384114|0DABB649161D071BD25E0E21F1268C342E661BB2|A7D73F8B5BD6D382431A294AC2FFB29360BE8D02|C808A0637C0FAA19D39D81E55AFDE7312183088E|9881A05296BAAB85EC3BF92ED9879258D
9A7DD7F|5279903EEE1A7F1906B00D580558E26B46A23735|EF29CB1BDCFA50AF97609F6E68825C5E042F7C71|DD99B10FCEA56FE988A90AB101D486269FDBFFF8|9A4C7A4998B04DD61C81D4DA9A4DD5CAA5753377|32218C596C1E49
66DCD0622C127D56876E298840|755C25E991CA7ECB1605DFCE78C659D3EF288A41
/home/sri/fuse/mem1/p2p_test.txt-E023145E332ABF228AA36C712EDE9439D9229DC4|BAFC362450D28AE3A182A489A1C2B05F01804404|648844261999D21DD45895FF457309D8BB11C367
~
~
```

Steps of operation

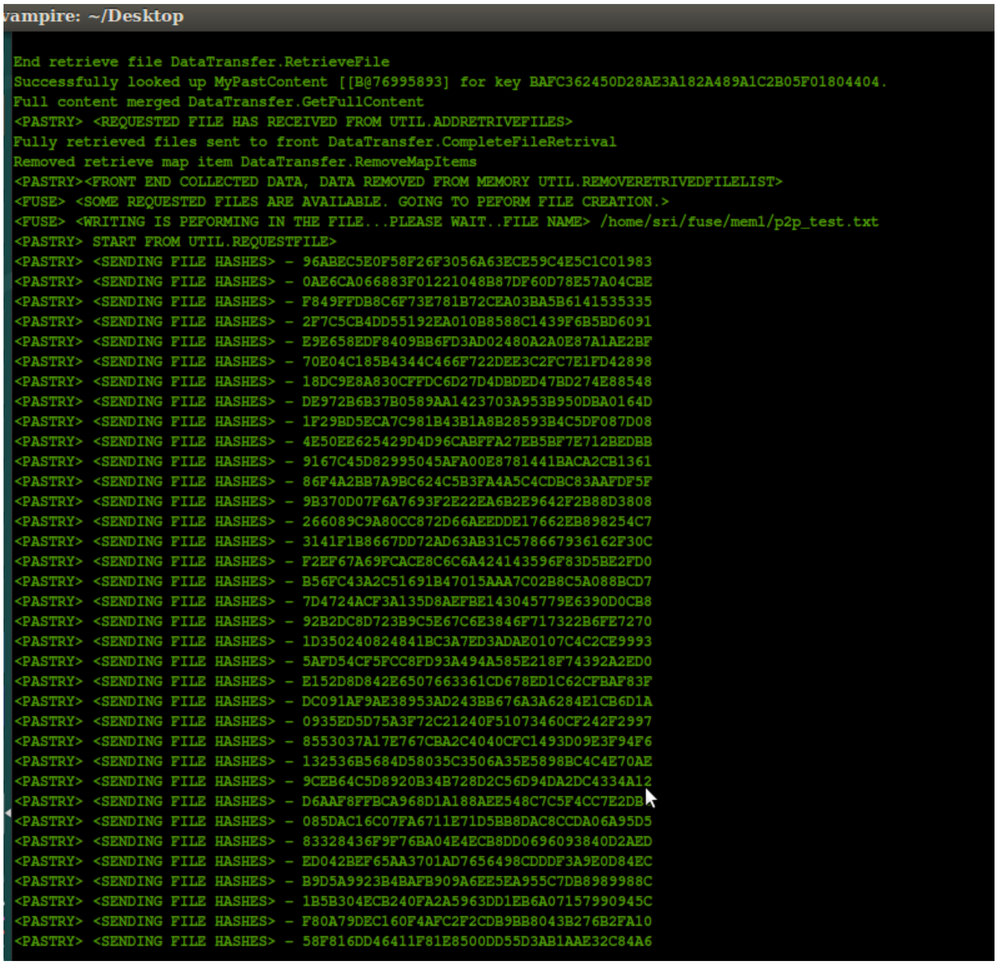
1. User executes the application and join the Pastry network. This will mount a FUSE file system for the user.
2. User can create files in the newly mounted file system. When user create and save the files, application will send the files, chunking them into pieces of pre-configured size to the Pastry DHT. Below image will show the console output provided when sending files.



```
MyApp 8C01C4F564D4D1E4B12E834D142A328931113B7A sending direct to [SNH: C9D1492A8
30792692F12D62E0CABAEC082C9ED5E//192.168.1.13:9006]
<PASTRY> STAT DETAIL - FILE COUNT> 4<TOTAL SIZE> 1360
<PASTRY> STAT DETAIL - FILE COUNT> 4<TOTAL SIZE> 1360
AdminMulticastContent.deliver([TOPIC C0985F6480E3C3F558F11D203D184D674DF7E9E6],MyScribeContent from [SNH: C9D1492A830792692F12D62E0CABAEC082C9ED5E//192.168.1.13:9006])
MyApp 8C01C4F564D4D1E4B12E834D142A328931113B7A sending direct to [SNH: C9D1492A830792692F12D62E0CABAEC082C9ED5E//192.168.1.13:9006]
<PASTRY> STAT DETAIL - FILE COUNT> 4<TOTAL SIZE> 1360
<PASTRY> STAT DETAIL - FILE COUNT> 4<TOTAL SIZE> 1360
<PASTRY> STAT DETAIL - FILE COUNT> 4<TOTAL SIZE> 1360
AdminMulticastContent.deliver([TOPIC C0985F6480E3C3F558F11D203D184D674DF7E9E6],MyScribeContent from [SNH: C9D1492A830792692F12D62E0CABAEC082C9ED5E//192.168.1.13:9006])
MyApp 8C01C4F564D4D1E4B12E834D142A328931113B7A sending direct to [SNH: C9D1492A830792692F12D62E0CABAEC082C9ED5E//192.168.1.13:9006]
<FUSE> <SWP AND SWX ARE CREATING HERE.REMOVE THEM > DONT INSERT THEM , IT WILL BE REMOVED AUTOMATICALLY <
<FUSE> <SWP AND SWX ARE CREATING HERE.REMOVE THEM > DONT INSERT THEM , IT WILL BE REMOVED AUTOMATICALLY <
<FUSE><FILE DELETE IS PERFORMING HERE> : /home/sri/fuse/mem1/.p2p_test.txt.swx
<FUSE><FILE DELETE IS PERFORMING HERE> : /home/sri/fuse/mem1/.p2p_test.txt.swp
<FUSE> <SWP AND SWX ARE CREATING HERE.REMOVE THEM > DONT INSERT THEM , IT WILL BE REMOVED AUTOMATICALLY <
<PASTRY> STAT DETAIL - FILE COUNT> 4<TOTAL SIZE> 1360
<FUSE> <FILE IS CREATED , NOW IT IS INSERTING IN TO MAP>:/home/sri/fuse/mem1/p2p_test.txt
<FUSE> <FILE SIZE HAS BEEN CHANGED , TIME TO SEND DATE TO PEERS> :/home/sri/fuse/mem1/p2p_test.txt
THE FILE IS GOING TO SPLIT - FILE SIZE - 55FILE NAME:/home/sri/fuse/mem1/p2p_test.txt
<PASTRY> START FROM UTIL.SENDDATA>
Start send data DataTransfer.SendData
Inserting MyPastContent [[B@b8d708f] at node [SNH: 8C01C4F564D4D1E4B12E834D142A328931113B7A//192.168.1.13:9009]
<FUSE><FILE DELETE IS PERFORMING HERE> : /home/sri/fuse/mem1/.p2p_test.txt.swp
<PASTRY> END FROM UTIL.SENDDATA>
MyPastContent [[B@b8d708f] successfully stored at 3 locations.
End send data DataTransfer.SendData
<PASTRY> START FROM UTIL.SENDDATA>
Start send data DataTransfer.SendData
Inserting MyPastContent [[B@b8d708f] at node [SNH: 8C01C4F564D4D1E4B12E834D142A328931113B7A//192.168.1.13:9009]
<PASTRY> END FROM UTIL.SENDDATA>
MyPastContent [[B@b8d708f] successfully stored at   locations.
End send data DataTransfer.SendData
<PASTRY> START FROM UTIL.SENDDATA>
Start send data DataTransfer.SendData
Inserting MyPastContent [[B@b8d708f] at node [SNH: 8C01C4F564D4D1E4B12E834D142A328931113B7A//192.168.1.13:9009]
<PASTRY> END FROM UTIL.SENDDATA>
File hash value :648844261999D21DD45895FF457309D8BB11C367

MyPastContent [[B@b8d708f] successfully stored at 3 locations.
End send data DataTransfer.SendData
NEW META INFO IS WRITING NOW  :/home/sri/emdc/p2p/p2p1/meta.ini
UPDATING THE META FILE ------ <FILE PATH>=HASHES|....:/home/sri/fuse/mem1/p2p_test.txt-E023145E332ABF228AA36C712EDE9439D9229DC4|BAFC362450D28AE3A182A489A1C2B05F01804404|648844261999D21DD4
5895FF457309D8BB11C367
<PASTRY> STAT DETAIL - FILE COUNT> 5<TOTAL SIZE> 1420
```

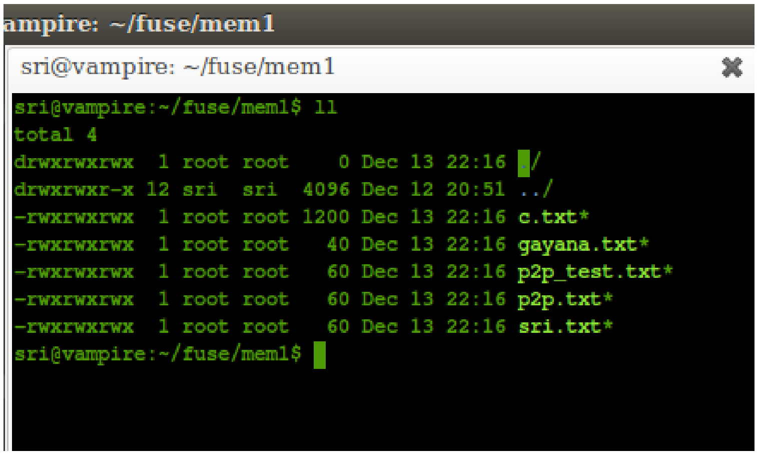
3. If the user update the file and save again new Hash values will be generated and above step 2 will be proceeded again.
4. When a previous user joined the network again, application will request the files in the metafile from the peers. Below figure shows the console output for the file request at start up.



vampire: ~/Desktop

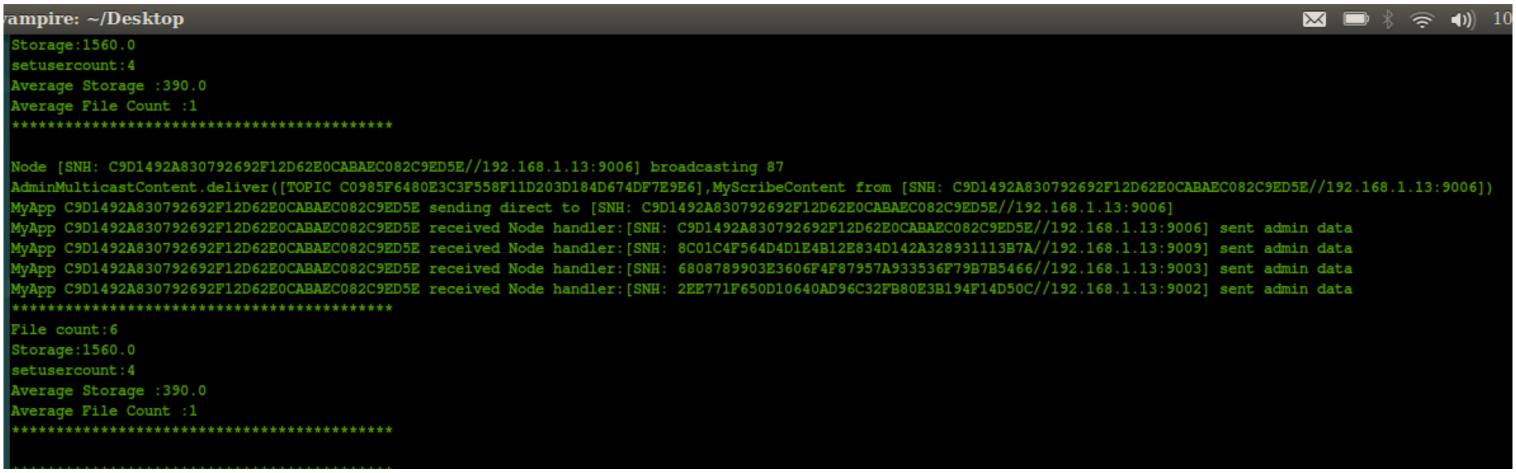
```
End retrieve file DataTransfer.RetrieveFile
Successfully looked up MyPastContent [[B@76995893] for key BAFC362450D28AE3A182A489A1C2B05F01804404.
Full content merged DataTransfer.GetFullContent
<PASTRY> <REQUESTED FILE HAS RECEIVED FROM UTIL.ADDRETRIVEFILES>
Fully retrieved files sent to front DataTransfer.CompleteFileRetrival
Removed retrieve map item DataTransfer.RemoveMapItems
<PASTRY><FRONT END COLLECTED DATA, DATA REMOVED FROM MEMORY UTIL.REMOVERETRIVEDFILELIST>
<FUSE> <SOME REQUESTED FILES ARE AVAILABLE. GOING TO PEFORM FILE CREATION.>
<FUSE> <WRITING IS PEFORMING IN THE FILE...PLEASE WAIT..FILE NAME> /home/sri/fuse/mem1/p2p_test.txt
<PASTRY> START FROM UTIL.REQUESTFILE>
<PASTRY> <SENDING FILE HASHES> - 96ABEC5E0F58F26F3056A63ECE59C4E5C1C01983
<PASTRY> <SENDING FILE HASHES> - 0AE6CA066883F01221048B87DF60D78E57A04CBE
<PASTRY> <SENDING FILE HASHES> - F849FFDB8C6F73E781B72CEA03BA5B6141535335
<PASTRY> <SENDING FILE HASHES> - 2F7C5CB4DD55192EA010B8588C1439F6B5BD6091
<PASTRY> <SENDING FILE HASHES> - E9E658EDF8409BB6FD3AD02480A2A0E87A1AE2BF
<PASTRY> <SENDING FILE HASHES> - 70E04C185B4344C466F722DEE3C2FC7E1FD42898
<PASTRY> <SENDING FILE HASHES> - 18DC9E8A830CFFDC6D27D4DBDED47BD274E88548
<PASTRY> <SENDING FILE HASHES> - DE972B6B37B0589AA1423703A953B950DBA0164D
<PASTRY> <SENDING FILE HASHES> - 1F29BD5ECA7C981B43B1A8B28593B4C5DF087D08
<PASTRY> <SENDING FILE HASHES> - 4E50EE625429D4D96CABFFA27EB5BF7E712BEDBB
<PASTRY> <SENDING FILE HASHES> - 9167C45D82995045AFA00E8781441BACA2CB1361
<PASTRY> <SENDING FILE HASHES> - 86F4A2BB7A9BC624C5B3FA4A5C4CDBC83AAFDF5F
<PASTRY> <SENDING FILE HASHES> - 9B370D07F6A7693F2E22EA6B2E9642F2B88D3808
<PASTRY> <SENDING FILE HASHES> - 266089C9A80CC872D66AEEDDE17662EB898254C7
<PASTRY> <SENDING FILE HASHES> - 3141F1B8667DD72AD63AB31C578667936162F30C
<PASTRY> <SENDING FILE HASHES> - F2EF67A69FCACE8C6C6A424143596F83D5BE2FD0
<PASTRY> <SENDING FILE HASHES> - B56FC43A2C51691B47015AAA7C02B8C5A088BCD7
<PASTRY> <SENDING FILE HASHES> - 7D4724ACF3A135D8AEFBE14304577 9E6390D0CB8
<PASTRY> <SENDING FILE HASHES> - 92B2DC8D723B9C5E67C6E3846F717322B6FE7270
<PASTRY> <SENDING FILE HASHES> - 1D350240824841BC3A7ED3ADAE0107C4C2CE9993
<PASTRY> <SENDING FILE HASHES> - 5AFD54CF5FCC8FD93A494A585E218F74392A2ED0
<PASTRY> <SENDING FILE HASHES> - E152D8D842E6507663361CD678ED1C62CFBAF83F
<PASTRY> <SENDING FILE HASHES> - DC091AF9AE38953AD243BB676A3A6284E1CB6D1A
<PASTRY> <SENDING FILE HASHES> - 0935ED5D75A3F72C21240F51073460CF242F2997
<PASTRY> <SENDING FILE HASHES> - 8553037A17E767CBA2C4040CFC1493D09E3F94F6
<PASTRY> <SENDING FILE HASHES> - 132536B5684D58035C3506A35E5898BC4C4E70AE
<PASTRY> <SENDING FILE HASHES> - 9CEB64C5D8920B34B728D2C56D94DA2DC4334A12
<PASTRY> <SENDING FILE HASHES> - D6AAF8FFBCA968D1A188AEE548C7C5F4CC7E2DB
<PASTRY> <SENDING FILE HASHES> - 085DAC16C07FA6711E71D5BB8DAC8CCDA06A95D5
<PASTRY> <SENDING FILE HASHES> - 83328436F9F76BA04E4ECB8DD0696093840D2AED
<PASTRY> <SENDING FILE HASHES> - ED042BEF65AA3701AD7656498CDDDF3A9E0D84EC
<PASTRY> <SENDING FILE HASHES> - B9D5A9923B4BAFB909A6EE5EA955C7DB8989988C
<PASTRY> <SENDING FILE HASHES> - 1B5B304ECB240FA2A5963DD1EB6A07157990945C
<PASTRY> <SENDING FILE HASHES> - F80A79DEC160F4AFC2F2CDB9BB8043B276B2FA10
<PASTRY> <SENDING FILE HASHES> - 58F816DD46411F81E8500DD55D3AB1AAE32C84A6
```

5. After the retrieval of files from peers, user can see the files mounted in the FUSE file system. Below image shows how user see the file system.



```
sri@vampire:~/fuse/mem1$ ll
total 4
drwxrwxrwx  1 root root    0 Dec 13 22:16 ./
drwxrwxr-x 12 sri  sri  4096 Dec 12 20:51 ../
-rwxrwxrwx  1 root root 1200 Dec 13 22:16 c.txt*
-rwxrwxrwx  1 root root   40 Dec 13 22:16 gayana.txt*
-rwxrwxrwx  1 root root   60 Dec 13 22:16 p2p_test.txt*
-rwxrwxrwx  1 root root   60 Dec 13 22:16 p2p.txt*
-rwxrwxrwx  1 root root   60 Dec 13 22:16 sri.txt*
sri@vampire:~/fuse/mem1$ 
```

6. The admin user can see the statistic information such as files shared in the network, active user count, total storage etc. Admin module will multicast messages time to time via the pastry network and collect above details. This will give approximate value for those information. Below image illustrates an admin view for information.



```
Storage:1560.0
setusercount:4
Average Storage :390.0
Average File Count :1
*******************************************

Node [SNH: C9D1492A830792692F12D62E0CABAEC082C9ED5E//192.168.1.13:9006] broadcasting 87
AdminMulticastContent.deliver([TOPIC C0985F6480E3C3F558F11D203D184D674DF7E9E6],MyScribeContent from [SNH: C9D1492A830792692F12D62E0CABAEC082C9ED5E//192.168.1.13:9006])
MyApp C9D1492A830792692F12D62E0CABAEC082C9ED5E sending direct to [SNH: C9D1492A830792692F12D62E0CABAEC082C9ED5E//192.168.1.13:9006]
MyApp C9D1492A830792692F12D62E0CABAEC082C9ED5E received Node handler:[SNH: C9D1492A830792692F12D62E0CABAEC082C9ED5E//192.168.1.13:9006] sent admin data
MyApp C9D1492A830792692F12D62E0CABAEC082C9ED5E received Node handler:[SNH: 8C01C4F564D4D1E4B12E834D142A328931113B7A//192.168.1.13:9009] sent admin data
MyApp C9D1492A830792692F12D62E0CABAEC082C9ED5E received Node handler:[SNH: 6808789903E3606F4F87957A933536F79B7B5466//192.168.1.13:9003] sent admin data
MyApp C9D1492A830792692F12D62E0CABAEC082C9ED5E received Node handler:[SNH: 2EE771F650D10640AD96C32FB80E3B194F14D50C//192.168.1.13:9002] sent admin data
*******************************************
File count:6
Storage:1560.0
setusercount:4
Average Storage :390.0
Average File Count :1
*******************************************
```

Test Observations

We generated 10 nodes with our application in a local newtwork and shared files in the network. We observed that the node joining consumed a considerable amount of time when the number of peers increased.

After joining the network the peers seemed to be communicating properly and in a stable manner. We removed upto 2 random nodes at a time but it did not make an impact to lose the files shared in the network. Once they rejoin they could retrieve all the files they shared earlier.

Note: To run the application we used below sample command line instruction

java -Djna.nosys=true -cp gs3FS.jar:lib/fusejan.jar:lib/fuse-jna-uber.jar:lib/Pastry_v1.jar:lib/xmlpull_1_1_3_4a.jar:lib/xpp3-1.1.3.4d_b2.jar

```
rice.uproject.implmeation.StartUp 9006 192.168.1.13 9002 /home/sri/fuse/mem1
/home/sri/emdc/p2p/p2p1
```

For admin console only use the below command line instruction.

```
 java -Djna.nosys=true -cp gs3FS.jar:lib/fusejan.jar:lib/fuse-jna-
uber.jar:lib/Pastry_v1.jar:lib/xmlpull_1_1_3_4a.jar:lib/xpp3-1.1.3.4d_b2.jar
rice.uproject.implmeation.StartUp 9006 192.168.1.13 9002 /home/sri/fuse/mem1
/home/sri/emdc/p2p/p2p1 1
```

For the testing we have used the multicast rerun duration as 25 seconds. When the network is large this value should be increased in to a considerably higher value in order to control the Pastry network congestion.

References

[1] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In Proc. IFIP/ACM Middleware, November 2001

[2] Antony Rowstron and Peter Druschel . Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility