

PROJECT Testing and Execution

Execution:

Simpleht.py has been modified, make sure the simpleht.py attached is used

1. Launch the servers:

`./launch_procs.py 2` or `python launch_procs.py 2`

Where `python launch_procs.py <num_servers>`

num_servers represents the number of servers within each server bank

2. **`python refresh_v1.py fusemount 'http://localhost:8000' 2`**

`Python refresh_v1.py <mount point> <start port http addr> <num of servers>`

Make sure the same number of servers is used in both `launch_procs.py` and `refresh_v1.py`. Run this in the directory where the fuse library is stored.

The ports are hardcoded in the code as port number 8000, 7000 and 9000. When you specify the number of servers, eg say 3: then port numbers 7000, 7001, 7002 , 8000, 8001, 8002, 9000, 9001, 9002 get executed.

3. To exit, use the `kill_servers.sh`

`./kill_servers.sh`

This will kill all the servers which are running.

Testing:

The following commands were tested for normal functioning of the file system

Regular File System Functions

Cd fusemount

Echo "test" > test

Cat test

Ls

Ln -s test file

Unlink file

Rm test

If the file is deleted before the symlink, then the symlink becomes red in color! Does not have any value !

Mkdir d

Echo "Hi" > hi

Mv hi li

Cp li ji

Ls -ltr

Touch hi

```
Chmod 0777 hi
Ls -ltr
```

Case 1: Server Down :

Kill any one server, or upto one cluster of servers (eg, 8000, 8001, 8002)

Access the contents in fusemount, the file system works using the majority voter.

Corrupting data in any of the remaining two servers, results in a failure

```
Ps ax
Kill <pid>      ##Kill the process which runs python simpleht.py --port = 8000 and not
xterm -e
```

Now return to the fusemount directory and type

```
ls          #Contents will be displayed
```

Case 2: Server UP:

Now get the servers up

```
python simpleht.py --port 8000
python simpleht.py --port 8001
```

Type ls in the fusemount directory or

Wait for 60s till the system refreshes

Accessing server directly

The contents of the server can be viewed by opening terminal and python interpreter

```
python
import xmlrpclib
from xmlrpclib import Binary
s = xmlrpclib.ServerProxy('http://localhost:8000')
s.print_content() #this lists the contents in the particular server window, the current
terminal will just display True
```

Corruption of Data in the file System:

Open a new terminal, keeping the file system running and corrupt the data by proxying into one of the servers

Open Terminal

```
python
import xmlrpclib
from xmlrpclib import Binary
s = xmlrpclib.ServerProxy('http://localhost:8000')
s.print_content() #this lists the contents in the particular server
From the print_content(), look for the file names and corrupt one of the files, or corrupt '/'
file, which will be present in 8000,7000,9000
s.put(Binary('/').Binary('Corruption'),1000)
```

Type ls in fusemount or wait for 60s until refresh is done.

When the server refreshes every 60s, or when the particular server is accessed, the contents will be restored from the remaining two copies

Any new file written into any server through ServerProxy will not be reflected in the File system even if it has a majority (Corruption in multiple servers)

Install iozone:

```
sudo apt-get install iozone3
```

```
cd fusemount
```

```
iozone -a -+u -n 4k -g 16k -b ../file.xls
```

SNAPSHOTS:

The descriptions are written below the figures.

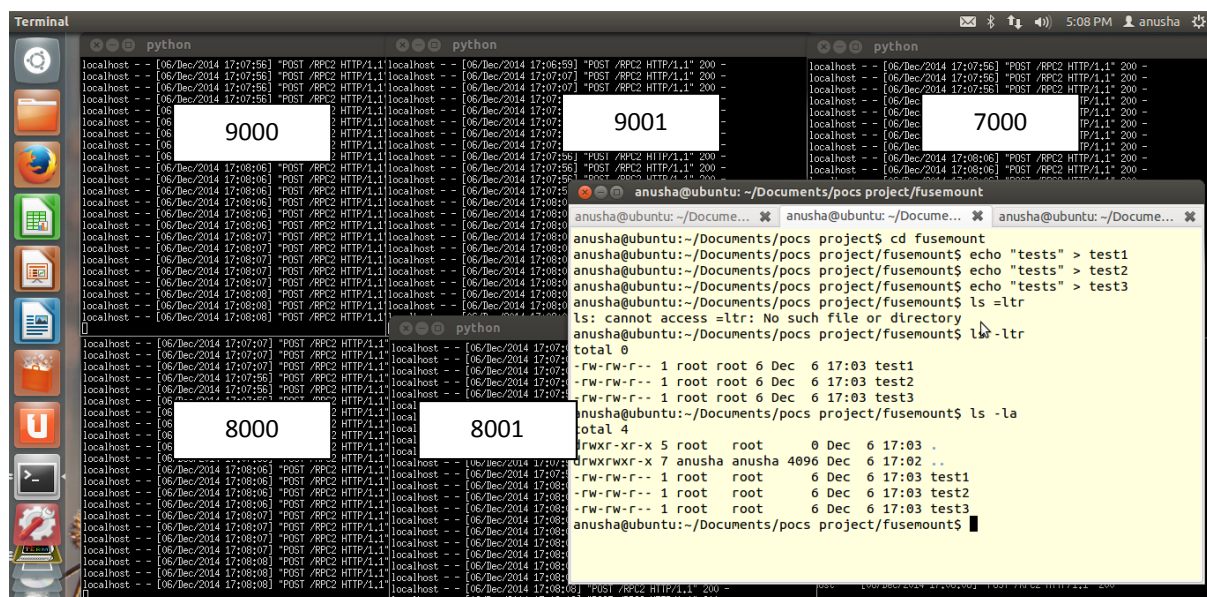


Fig1 : All 6 servers up and running.

Fig 1 displays all the 5 servers are running, and the mounted file system has 3 files saved in it.

The terminal window displays the output of the `ps` command, showing various processes running on the system. The processes include `bash`, `python`, `xterm`, `python simpleht.py`, `python refresh_v1.py`, and `ps ax`. The `python simpleht.py` processes are running on ports 8000, 8001, 7000, and 7001. The `python refresh_v1.py` process is running on port 8000. The `ps ax` command shows the system's process table.

```
anusha@ubuntu:~/Documents/pocs project/server$ ps ax
2811 pts/0 Ss 0:01 bash
2892 pts/2 Ss 0:01 bash
2948 pts/3 Ss+ 0:01 bash
3005 ? S 0:00 /usr/lib/deja-dup/deja-dup/deja-dup-monitor
3056 pts/11 Ss 0:00 bash
3113 pts/11 S+ 0:01 python
3896 ? S 0:00 /usr/lib/at-spi2-core/at-spi-bus-launcher
4769 ? S 0:00 [kworker/0:0]
4928 ? S 0:00 [kworker/u:0]
5536 ? S 0:00 [kworker/0:1]
5636 pts/2 S 0:00 xterm -e python simpleht.py --port=8000
5637 pts/2 S 0:00 xterm -e python simpleht.py --port=8001
5638 pts/2 S 0:00 xterm -e python simpleht.py --port=7000
5639 pts/2 S 0:00 xterm -e python simpleht.py --port=7001
5645 pts/7 Ss+ 0:00 python simpleht.py --port=8001
5646 pts/6 Ss+ 0:00 python simpleht.py --port=7000
5652 pts/9 Ss+ 0:00 python simpleht.py --port=7001
5654 pts/10 Ss+ 0:00 python simpleht.py --port=8000
5657 pts/0 S 0:04 python refresh_v1.py fusemount http://localhost:8000
5719 ? S 0:00 [kworker/0:2]
5775 pts/2 R+ 0:00 ps ax

anusha@ubuntu:~/Documents/pocs project/server$ ls -la
total 4
drwxr-xr-x 5 root root 0 Dec 6 17:03 .
drwxrwxr-x 7 anusha anusha 4096 Dec 6 17:02 ..
-rw-rw-r-- 1 root root 6 Dec 6 17:03 test1
-rw-rw-r-- 1 root root 6 Dec 6 17:03 test2
-rw-rw-r-- 1 root root 6 Dec 6 17:03 test3

anusha@ubuntu:~/Documents/pocs project/fusemount$
```

Fig 2: Both servers down

When one cluster of servers are down, the majority still exists and the file system functions normally.

The terminal window displays the output of the `ps` command, showing various processes running on the system. The processes include `bash`, `python`, `xterm`, `python simpleht.py`, `python refresh_v1.py`, and `ps ax`. The `python simpleht.py` processes are running on ports 8000, 8001, 7000, and 7001. The `python refresh_v1.py` process is running on port 8000. The `ps ax` command shows the system's process table.

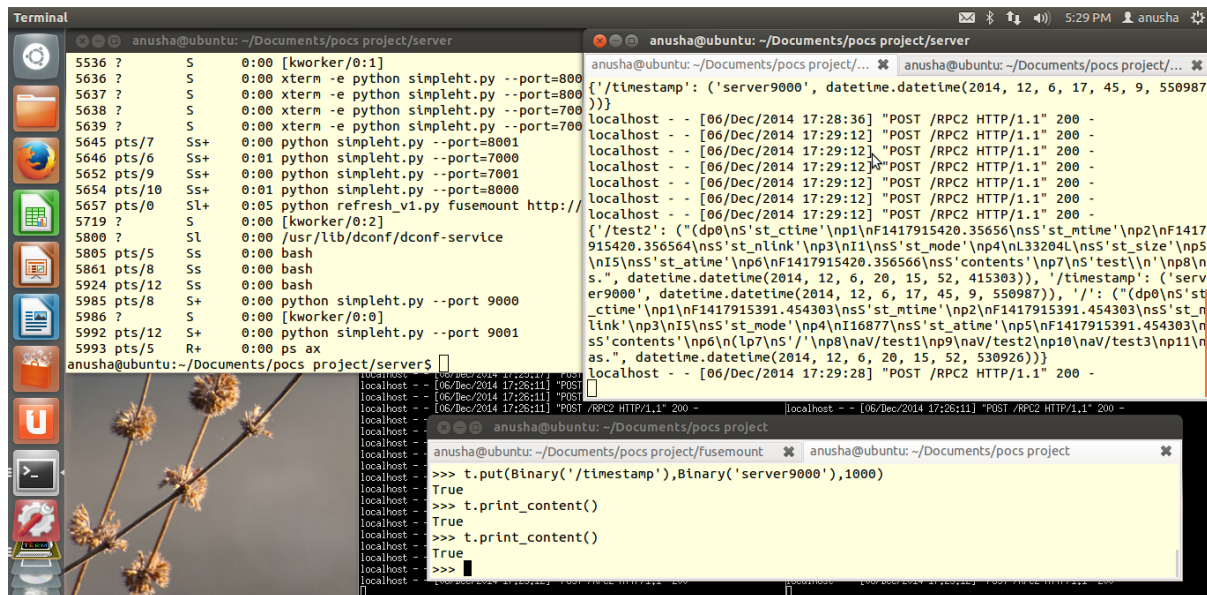
```
anusha@ubuntu:~/Documents/pocs project/server$ ps ax
5536 ? S 0:00 [kworker/0:1]
5636 ? S 0:00 xterm -e python simpleht.py --port=8000
5637 ? S 0:00 xterm -e python simpleht.py --port=8000
5638 ? S 0:00 xterm -e python simpleht.py --port=7000
5639 ? S 0:00 xterm -e python simpleht.py --port=7000
5645 pts/7 Ss+ 0:00 python simpleht.py --port=8001
5646 pts/6 Ss+ 0:01 python simpleht.py --port=7000
5652 pts/9 Ss+ 0:00 python simpleht.py --port=7001
5654 pts/10 Ss+ 0:01 python simpleht.py --port=8000
5657 pts/0 S 0:05 python refresh_v1.py fusemount http://
5719 ? S 0:00 [kworker/0:2]
5800 ? S 0:00 /usr/lib/dconf/dconf-service
5805 pts/5 Ss 0:00 bash
5861 pts/8 Ss 0:00 bash
5924 pts/12 Ss 0:00 bash
5985 pts/8 S+ 0:00 python simpleht.py --port 9000
5986 ? S 0:00 [kworker/0:0]
5992 pts/12 S+ 0:00 python simpleht.py --port 9001
5993 pts/5 R+ 0:00 ps ax

anusha@ubuntu:~/Documents/pocs project/server$ python simpleht.py --port 9000
[
  localhost -- [06/Dec/2014 17:15:34] "POST /RPC2 HTTP/1.1" 200 -
  localhost -- [06/Dec/2014 17:16:32] "POST /RPC2 HTTP/1.1" 200 -
  {'timestamp': ('hdfd', datetime.datetime(2014, 12, 6, 17, 33, 12, 953890))}
  localhost -- [06/Dec/2014 17:16:37] "POST /RPC2 HTTP/1.1" 200 -
]

anusha@ubuntu:~/Documents/pocs project/fusemount$
```

Fig 3: the servers 9000 and 9001 are up

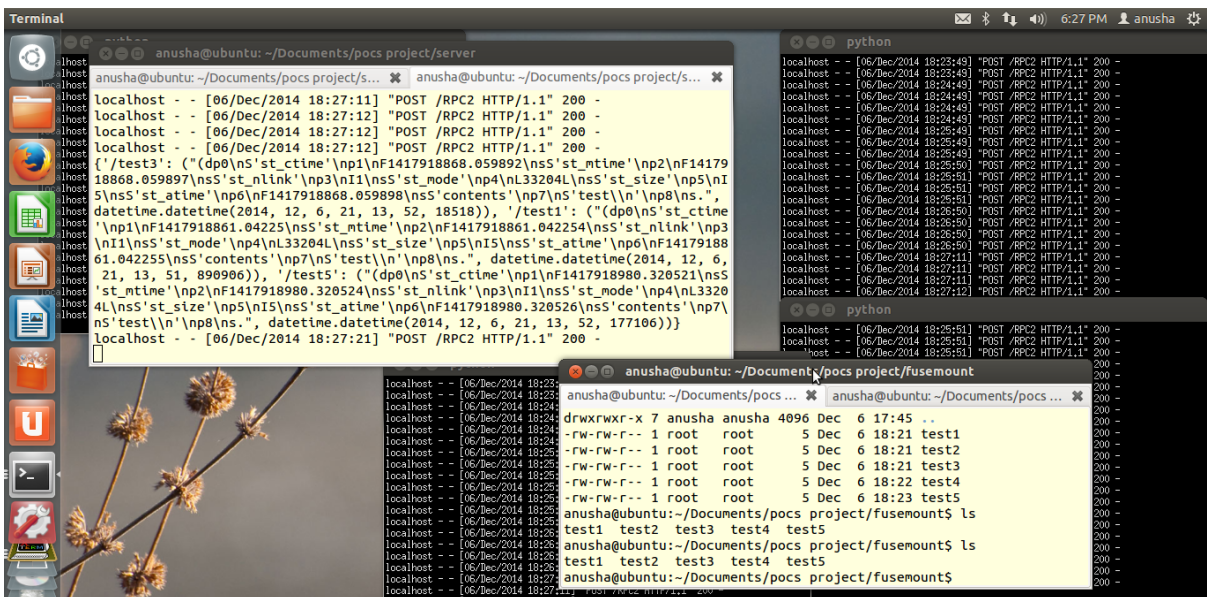
The servers 9000 and 9001 are up and have no files in them. Either a file access or a periodic refresh restores the contents of the servers. In this case, the servers are restored by the refresh function. The figure shows an external entry added to create a time stamp when the server is pulled back up. The next image displays the data after refresh gets executed. When this is updated the file metadata is maintained to be same as the metadata on the other servers, i.e. it does not get altered.



The terminal shows a user named 'anusha' at 'anusha@ubuntu: ~/Documents/pocs project/server'. The left pane displays a list of processes running on various hosts, including 'xterm', 'python', and 'ps ax'. The right pane shows a large block of JSON data representing server state, including timestamps, server IDs, and content links. The bottom pane shows a series of 'True' responses from a script, indicating successful refreshes.

Fig 4 Servers after-Refresh

The servers 9000 and 9001 are refreshed and their contents are brought back up using the two unaltered copies which exist in two other server clusters(namely 8000 and 7000). The refresh also preserves any additional data that was written to the server (the timestamp entry in this example).



The terminal shows a user named 'anusha' at 'anusha@ubuntu: ~/Documents/pocs project/server'. The left pane displays a list of processes running on various hosts, including 'xterm', 'python', and 'ps ax'. The right pane shows a large block of JSON data representing server state, including timestamps, server IDs, and content links. The bottom pane shows a series of 'True' responses from a script, indicating successful refreshes.

Fig 5 restore through access

Here, the contents of the server are restored through file access in the file system. When a file is accessed, two correct copies will exist in two servers. The status of the third server is checked and if it is up, the contents will be written on it.


```

anusha@ubuntu: ~/Documents/pocs project/server
5536 ? S 0:00 [kworker/0:1]
5636 ? S 0:00 xterm -e python simpleht.py --port=8000
5637 ? S 0:00 xterm -e python simpleht.py --port=8000
5638 ? S 0:00 xterm -e python simpleht.py --port=7000
5639 ? S 0:00 xterm -e python simpleht.py --port=7000
5645 pts/7 Ss+ 0:00 python simpleht.py --port=8001
5646 pts/6 Ss+ 0:01 python simpleht.py --port=7000
5652 pts/9 Ss+ 0:00 python simpleht.py --port=7001
5654 pts/10 Ss+ 0:01 python simpleht.py --port=8000
5657 pts/0 SL+ 0:05 python refresh_v1.py fusemount http://
5719 ? S 0:00 [kworker/0:2]
5800 ? SL 0:00 /usr/lib/dconf/dconf-service
5805 pts/5 Ss 0:00 bash
5861 pts/8 Ss 0:00 bash
5924 pts/12 Ss 0:00 bash
5985 pts/8 S+ 0:00 python simpleht.py --port 9000
5986 ? S 0:00 [kworker/0:0]
5992 pts/12 S+ 0:00 python simpleht.py --port 9001
5993 pts/5 R+ 0:00 ps ax

anusha@ubuntu:~/Documents/pocs project/server$
localhost - - [06/Dec/2014 17:29:28] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:29:28] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:29:28] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:30:12] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:30:12] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:30:12] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:30:12] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:30:12] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:30:34] "POST /RPC2 HTTP/1.1" 200 -
{'/test2': ('CORRUPTED', datetime.datetime(2014, 12, 6, 17, 47, 14, 404154)), '/timestamp': ('server9000', datetime.datetime(2014, 12, 6, 17, 45, 9, 550987)), '/': ('(dp0\N5'st_ctime'\np1\F1417915391.454303\N5'st_mtlme'\np2\F1417915391.454303\N5'st_nlink'\np3\N5\N5'st_mode'\np4\N16877\N5'st_atime'\np5\F1417915391.454303\N5'st_contents'\np6\N(lp7\N5')\np8\N5\test1\np9\N5\test2\np10\N5\test3\np11\N5', datetime.datetime(2014, 12, 6, 20, 15, 52, 530926))}}
localhost - - [06/Dec/2014 17:30:35] "POST /RPC2 HTTP/1.1" 200 -
anusha@ubuntu:~/Documents/pocs project$
anusha@ubuntu:~/Documents/pocs project/fusemount$
>>> t.print_content()
True
>>> t.put(Binary('/test2'), Binary('CORRUPTED'), 1000)
True
>>> t.print_content()
True
>>>

```

Fig 6 Corruption of a file in server 9000

A file '/test2' in server 9000 is corrupted. The contents and metadata of the server is corrupted. The corruption is restored either through file access or refresh.

```

anusha@ubuntu: ~/Documents/pocs project/server
as.", datetime.datetime(2014, 12, 6, 20, 15, 52, 530926))}}
localhost - - [06/Dec/2014 17:29:28] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:30:12] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:30:12] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:30:12] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:30:12] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:30:12] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:30:34] "POST /RPC2 HTTP/1.1" 200 -
{'/test2': ('CORRUPTED', datetime.datetime(2014, 12, 6, 17, 47, 14, 404154)), '/timestamp': ('server9000', datetime.datetime(2014, 12, 6, 17, 45, 9, 550987)), '/': ('(dp0\N5'st_ctime'\np1\F1417915391.454303\N5'st_mtlme'\np2\F1417915391.454303\N5'st_nlink'\np3\N5\N5'st_mode'\np4\N16877\N5'st_atime'\np5\F1417915391.454303\N5'st_contents'\np6\N(lp7\N5')\np8\N5\test1\np9\N5\test2\np10\N5\test3\np11\N5', datetime.datetime(2014, 12, 6, 20, 15, 52, 530926))}}
localhost - - [06/Dec/2014 17:30:35] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:31:12] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:31:12] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:31:12] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:31:12] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:31:12] "POST /RPC2 HTTP/1.1" 200 -
localhost - - [06/Dec/2014 17:31:12] "POST /RPC2 HTTP/1.1" 200 -
{'/test2': ('(dp0\N5'st_ctime'\np1\F1417915420.35656\N5'st_mtime'\np2\F1417915420.35656\N5'st_nlink'\np3\N1\N5'st_mode'\np4\N33204\N5'st_size'\np5\N5\N5'st_atime'\np6\F1417915420.35656\N5'st_contents'\np7\N5\test1\np8\N5', datetime.datetime(2014, 12, 6, 20, 17, 52, 875604)), '/timestamp': ('server9000', datetime.datetime(2014, 12, 6, 17, 45, 9, 550987)), '/': ('(dp0\N5'st_ctime'\np1\F1417915391.454303\N5'st_mtime'\np2\F1417915391.454303\N5'st_nlink'\np3\N5\N5'st_mode'\np4\N16877\N5'st_atime'\np5\F1417915391.454303\N5'st_contents'\np6\N(lp7\N5')\np8\N5\test1\np9\N5\test2\np10\N5\test3\np11\N5', datetime.datetime(2014, 12, 6, 20, 15, 52, 530926))}}
localhost - - [06/Dec/2014 17:31:29] "POST /RPC2 HTTP/1.1" 200 -

```

Fig 7 Corrupted data restored

The file '/test2' which was corrupted is restored through the refresh function.