Hashy

Experience with DHT with kademlia library in Python 3.

Requirements

- Tested on Mac OS only, this can be run on Linux and Window system with some modification.
- `Python 3` install HERE
  - Use `python --version` to make sure your python is version 3.0+
- `Kademlia` : A distributed hash table designed for decentralized peer-to-peer computer networks.
- `Asyncio` : Facilitates keeping the program running asynchronously.
- `Aioconsole` : Enables asynchronous input handling.
- `Pillow` : Library for image encoding and decoding.
- `Lorem-text` : Library for generating testing strings.

Data Usage to Measure

A set of images in folder `to_send` are picked with various sizes and types:

- 1_24kb.jpeg
- 2_360kb.png
- 3_2mb.jpg
- 4_10mb.png
- 5_15mb.png
- 6_18mb.png
- 7_19mb.png

A set of random lorem ipsum string are generated.

Measurement Design

- Duration is not included:
  - the time to encode, decode the image.
  - the time to generate strings.
  - the time to save csv, or save image.

Files, `RECORDS` folder

- **Data is manipulated into `report.xlsx` for the purpose of graphing, from:.**
  - `img_read.csv` : Image information such as image name, image size, image chunk from the network, and read/download duration are reported to this file.
  - `img_write.csv` : Image name, image size, number of chunks, average chunk size, number of bytes totally, and upload duration are reported to this file.
  - `str_read.csv` : key, length of value, and read duration are reported to this file.
  - `str_write.csv` : key, length of value, and set duration are reported to this file.

At the end of each table in `report.xlsx` , there are average duration.

Other files:

- `Records` folder have reports used to make this report. Contains `e1` , `e2` , `e3` folders for collected experience data.
- `Report` folder uses to save `.csv` files after recording duration.
- `download` folder uses to save download images from the networks.

Installation

1. Go to `Hashy` directory, use the following command to install all dependencies:

```
pip install -r requirements.txt
```

How to Run

To Start an initial node (I call it server)

In Hashy directory, use the following command:

```
python server.py <PORT>
```

- Where `<PORT>` is the unused PORT number that you want your server running on.

Example: `python server.py 5678`

To Start Other Nodes and Join the Network

In the Hashy directory, use the following command:

```
python peer.py <NETWORK_PORT> <PORT>
```

Commands

1. Pick action
    - Type `s` to set value to the network
    - Type `g` to get value from the network

    ```
    Set [s], Get [g], Clear Screen[c] or Quit [q]:
    ```

2. Pick type of value to set
    - Type `kv` if you want to set key and value pair to the network
    - Type `img` if you want to upload an image to the network

    ```
    Enter type <key,value> [kv], image [img], defined img [ai] lorem mode[l]:
    ```

If it's image, please enter full path to the image.

For example: `'/Users/USER_NAME/Hashy/to_send/7_19mb.png'`

```
Enter the file path [.jpg|.jpeg|.png]:
```

- Type `ai` to send a set of defined images in folder `to_send`
  - For testing purpose, Images in `to_send` must be kept ORIGINAL
- Type `l` to send 200 random lorem ipsum in various size
  - For testing purpose, key and value will not be displayed at run time
  - Use `g` to get desired key:value
    - For example, the key can be `lorem_0` , `lorem_5` ,... `lorem_1000`

3. Pick type of value to get

- Type `k` to get a value of a key.
- Type `img` to download the image with image name without extension
- Type `ai` to download the defined images that WAS UPLOADED to the networks
  - Download to the `download` folder in `Hashy` directory
- Type `l` to get 200 key:values that was SET to the netoworks
  - For the testing purpose, `l` mode will not show you the result.

```
Enter type key [k], image [img], defined img [ai], lorem mode [l]:
```

Where

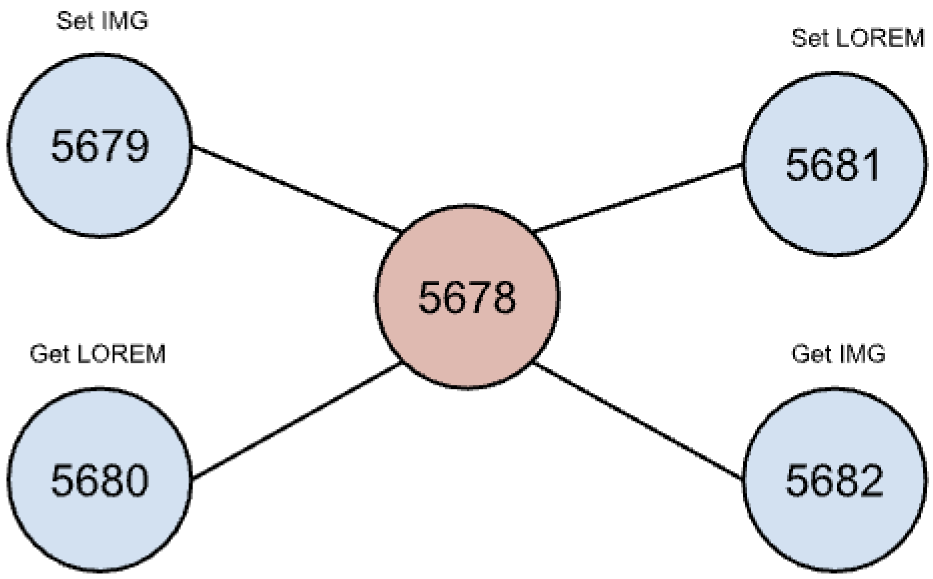
- `<NETWORK_PORT>` is the port of the server, or other running nodes in the network.
- `<PORT>` is the unused PORT number on your machine that you want your node running on.

Example: To connect to the initial network running on port `5678` and I have port `6000` free/unused on my machine.

- Run: `python peer.py 5678 6000`

Showcases

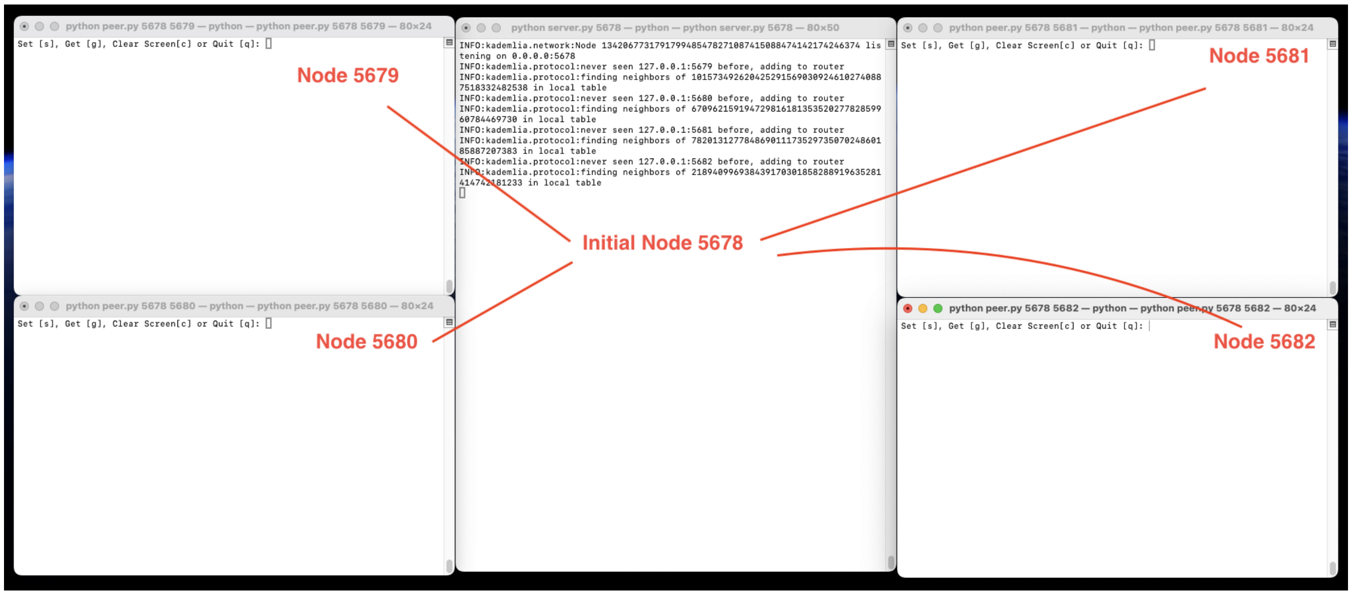
Experience 1



1. Start a Node at port `5678`
2. Start other Nodes at port `5679` , `5680` , `5681` , `5682` and connect them to the network (port `5678`)
3. Set a given Images to the network on port `5679` and set a random Lorem Text on port `5681` to the network.
4. On port `5682` , get a set of Images, and on port `5680` , get a set of Lorem Text from the network.

Steps

Create the network following Graph 1 above.


python peer.py 5678 5679 — python — python peer.py 5678 5679 — 80×24
python server.py 5678 — python — python server.py 5678 — 80×50
python peer.py 5678 5681 — python — python peer.py 5678 5681 — 80×24
python peer.py 5678 5680 — python — python peer.py 5678 5680 — 80×24
python peer.py 5678 5682 — python — python peer.py 5678 5682 — 80×24
Node 5679
Node 5681
Initial Node 5678
Node 5680
Node 5682

```
Set [s], Get [g], Clear Screen[c] or Quit [q]:
```

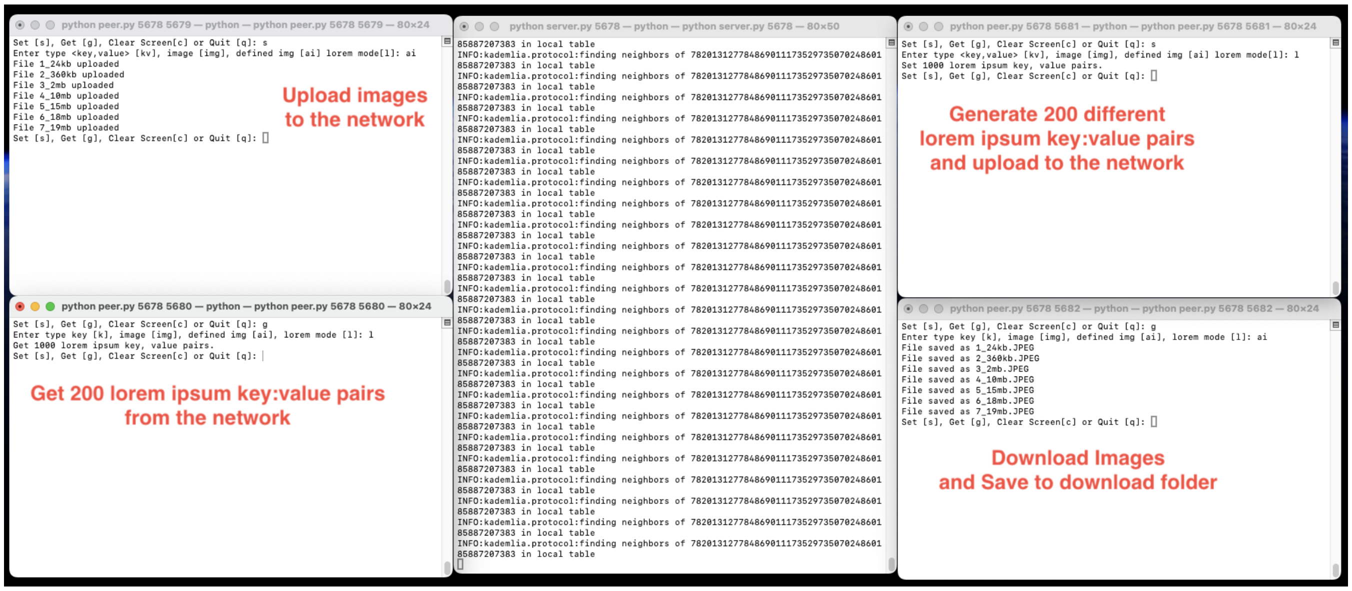
```
INFO:kademlia.network:Node 1342067731791799485478271087415088474142174246374 lis
tening on 0.0.0.0:5678
INFO:kademlia.protocol:never seen 127.0.0.1:5679 before, adding to router
INFO:kademlia.protocol:finding neighbors of 10157349262042529156903092461027408
8751833248253 in local table
INFO:kademlia.protocol:never seen 127.0.0.1:5680 before, adding to router
INFO:kademlia.protocol:finding neighbors of 67096215919472981618135352027782859
960784469730 in local table
INFO:kademlia.protocol:never seen 127.0.0.1:5681 before, adding to router
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:never seen 127.0.0.1:5682 before, adding to router
INFO:kademlia.protocol:finding neighbors of 21894099693843917030185828891963528141474218123 in local table
```

```
Set [s], Get [g], Clear Screen[c] or Quit [q]:
```

```
Set [s], Get [g], Clear Screen[c] or Quit [q]:
```

```
Set [s], Get [g], Clear Screen[c] or Quit [q]:
```

Upload images and lorem ipsum strings.


python peer.py 5678 5679 — python — python peer.py 5678 5679 — 80×24
python server.py 5678 — python — python server.py 5678 — 80×50
python peer.py 5678 5681 — python — python peer.py 5678 5681 — 80×24
python peer.py 5678 5680 — python — python peer.py 5678 5680 — 80×24
python peer.py 5678 5682 — python — python peer.py 5678 5682 — 80×24
Upload images
to the network
Generate 200 different
lorem ipsum key:value pairs
and upload to the network
Get 200 lorem ipsum key:value pairs
from the network
Download Images
and Save to download folder

```
Set [s], Get [g], Clear Screen[c] or Quit [q]: s
Enter type <key,value> [kv], image [img], defined img [ai] lorem mode[l]: ai
File 1_24kb uploaded
File 2_360kb uploaded
File 3_2mb uploaded
File 4_10mb uploaded
File 5_15mb uploaded
File 6_18mb uploaded
File 7_19mb uploaded
Set [s], Get [g], Clear Screen[c] or Quit [q]:
```

```
85887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
INFO:kademlia.protocol:finding neighbors of 78201312778486901117352973507024860185887207383 in local table
```

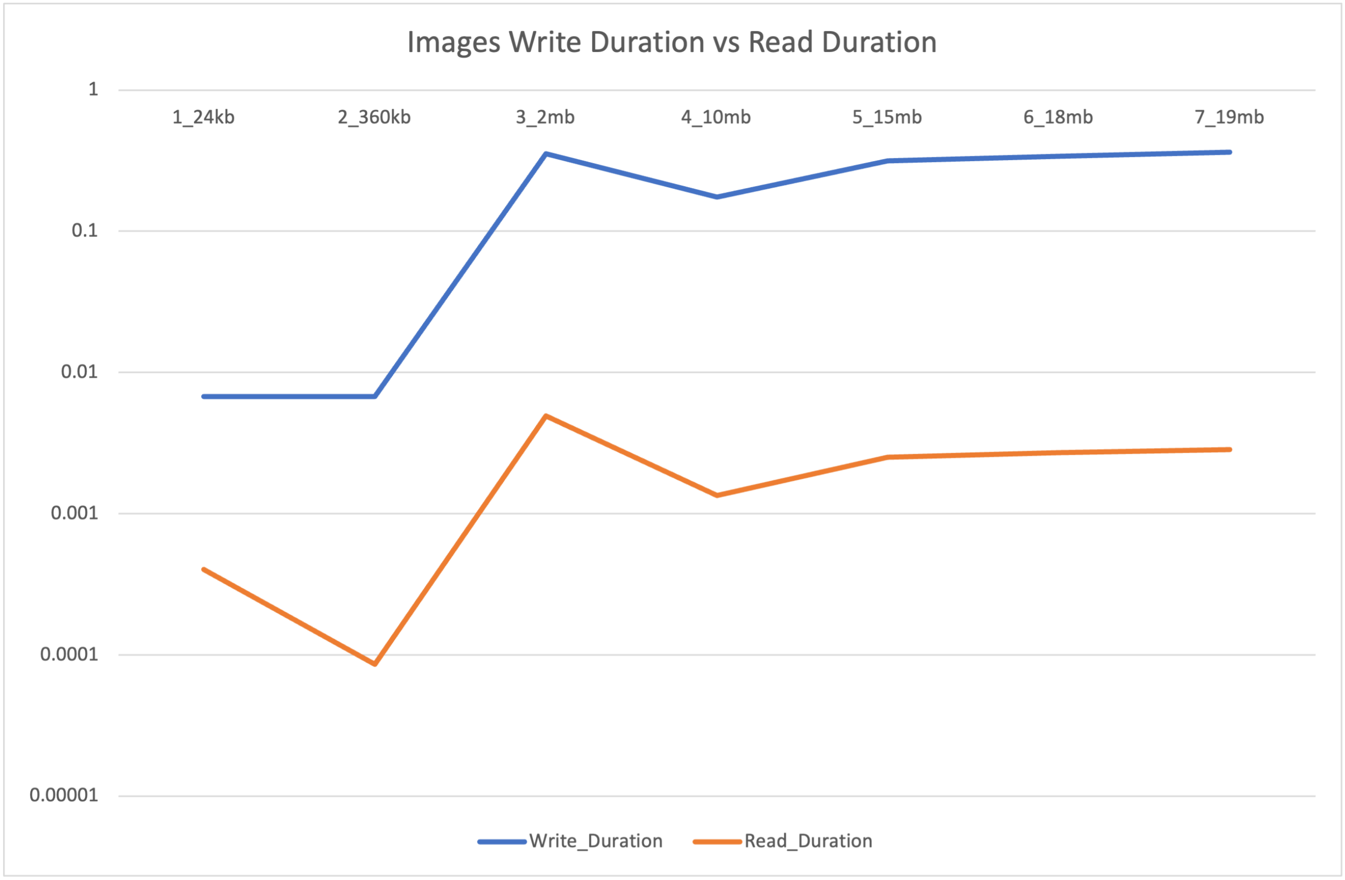
```
Set [s], Get [g], Clear Screen[c] or Quit [q]: s
Enter type <key,value> [kv], image [img], defined img [ai] lorem mode[l]: l
Set 1000 lorem ipsum key, value pairs.
Set [s], Get [g], Clear Screen[c] or Quit [q]:
```

```
Set [s], Get [g], Clear Screen[c] or Quit [q]: g
Enter type key [k], image [img], defined img [ai], lorem mode [l]: l
Get 1000 lorem ipsum key, value pairs.
Set [s], Get [g], Clear Screen[c] or Quit [q]:
```

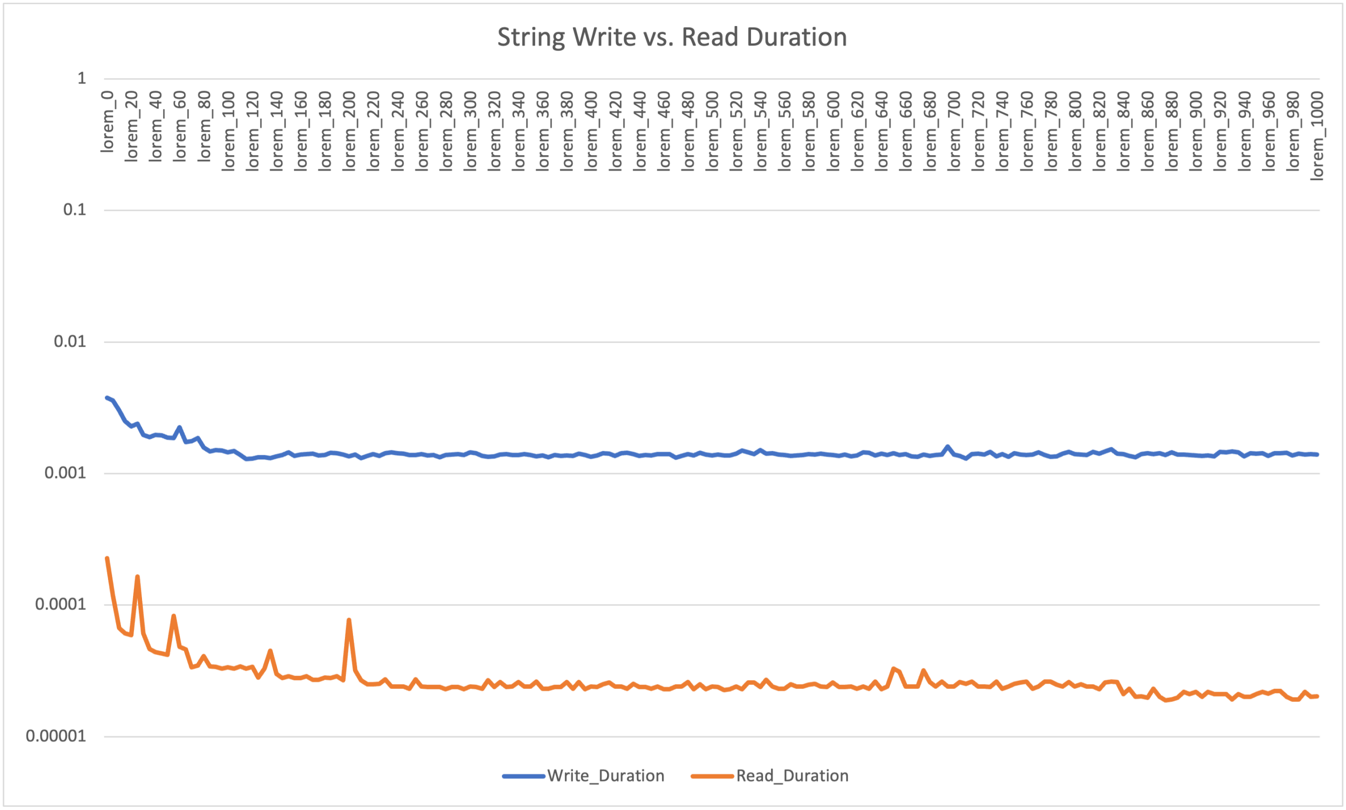
```
Set [s], Get [g], Clear Screen[c] or Quit [q]: g
Enter type key [k], image [img], defined img [ai], lorem mode [l]: ai
File saved as 1_24kb.JPEG
File saved as 2_360kb.JPEG
File saved as 3_2mb.JPEG
File saved as 4_10mb.JPEG
File saved as 5_15mb.JPEG
File saved as 6_18mb.JPEG
File saved as 7_19mb.JPEG
Set [s], Get [g], Clear Screen[c] or Quit [q]:
```

Measurements:

- Graph 1: Write and Read Images Duration



- Graph 2: Duration of set and get 200 lorem ipsum, each string increase by 5 words

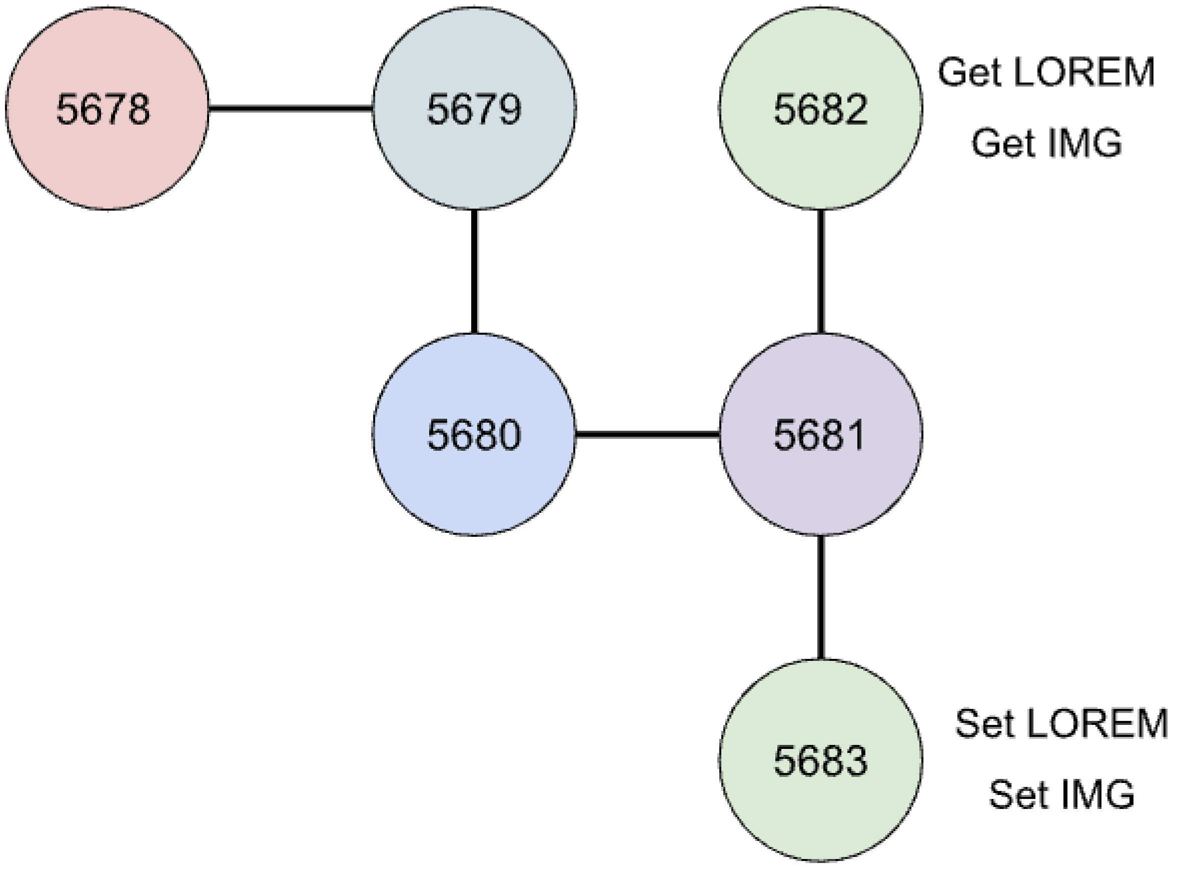


Discussion

Purpose of experience 1 is to explore how the DHT network works by setting up a central node, and let other peers connect to the central node.

- In Graph 1, Write duration is significantly slower than read duration (Images).
  - Write an image to DHT system is about `221 ms` slower than read an image.
- In Graph 2, Write duration is also significantly slower than read duration (Strings).
  - Write a string to DHT system is about `1.44 ms` slower than read a string.

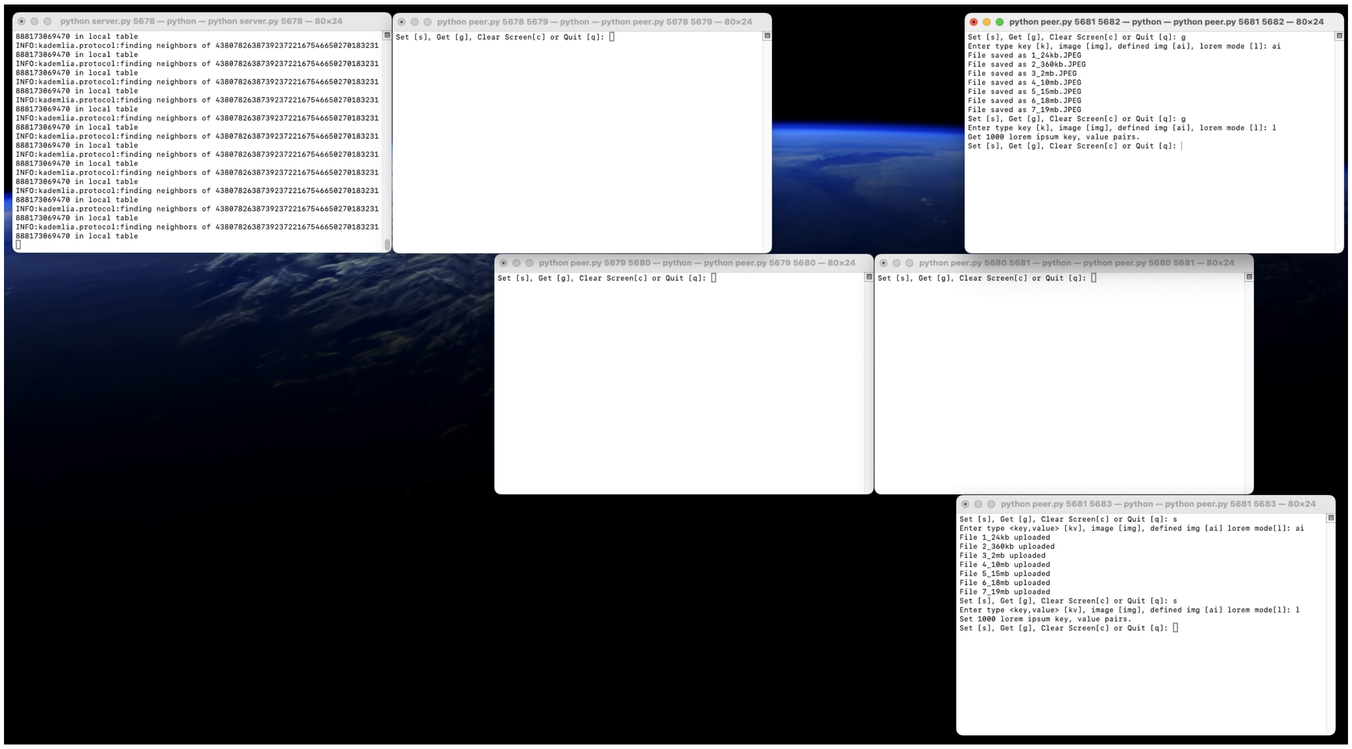
Experience 2



Steps

1. Start a Node at port `5678`

2. Start Node `5679` and connect to `5678`
3. Start Node `5680` and connect to `5679`
4. Start Node `5681` and connect to `5680`
5. Start Node `5682` and Connect to `5681`
6. Start Node `5683` and Connect to `5681`
7. Set Images and Lorem from Node `5683` to the network
8. Get Images and Lorem at Node `5682`


python server.py 5678 — python — python server.py 5678 — 80×24
python peer.py 5678 5679 — python — python peer.py 5678 5679 — 80×24
python peer.py 5681 5682 — python — python peer.py 5681 5682 — 80×24
python peer.py 5679 5680 — python — python peer.py 5679 5680 — 80×24
python peer.py 5680 5681 — python — python peer.py 5680 5681 — 80×24
python peer.py 5681 5683 — python — python peer.py 5681 5683 — 80×24

```
888173069470 in local table
INFO:kademlia.protocol:finding neighbors of 438078263873923722167546650270183231
888173069470 in local table
INFO:kademlia.protocol:finding neighbors of 438078263873923722167546650270183231
888173069470 in local table
INFO:kademlia.protocol:finding neighbors of 438078263873923722167546650270183231
888173069470 in local table
INFO:kademlia.protocol:finding neighbors of 438078263873923722167546650270183231
888173069470 in local table
INFO:kademlia.protocol:finding neighbors of 438078263873923722167546650270183231
888173069470 in local table
INFO:kademlia.protocol:finding neighbors of 438078263873923722167546650270183231
888173069470 in local table
INFO:kademlia.protocol:finding neighbors of 438078263873923722167546650270183231
888173069470 in local table
INFO:kademlia.protocol:finding neighbors of 438078263873923722167546650270183231
888173069470 in local table
INFO:kademlia.protocol:finding neighbors of 438078263873923722167546650270183231
888173069470 in local table
INFO:kademlia.protocol:finding neighbors of 438078263873923722167546650270183231
888173069470 in local table
INFO:kademlia.protocol:finding neighbors of 438078263873923722167546650270183231
888173069470 in local table
```

```
Set [s], Get [g], Clear Screen[c] or Quit [q]:
```

```
Set [s], Get [g], Clear Screen[c] or Quit [q]: g
Enter type key [k], image [img], defined img [ai], lorem mode [l]: ai
File saved as 1_24kb.JPEG
File saved as 2_360kb.JPEG
File saved as 3_2mb.JPEG
File saved as 4_10mb.JPEG
File saved as 5_15mb.JPEG
File saved as 6_18mb.JPEG
File saved as 7_19mb.JPEG
Set [s], Get [g], Clear Screen[c] or Quit [q]: g
Enter type key [k], image [img], defined img [ai], lorem mode [l]: l
Get 1000 lorem ipsum key, value pairs.
Set [s], Get [g], Clear Screen[c] or Quit [q]:
```

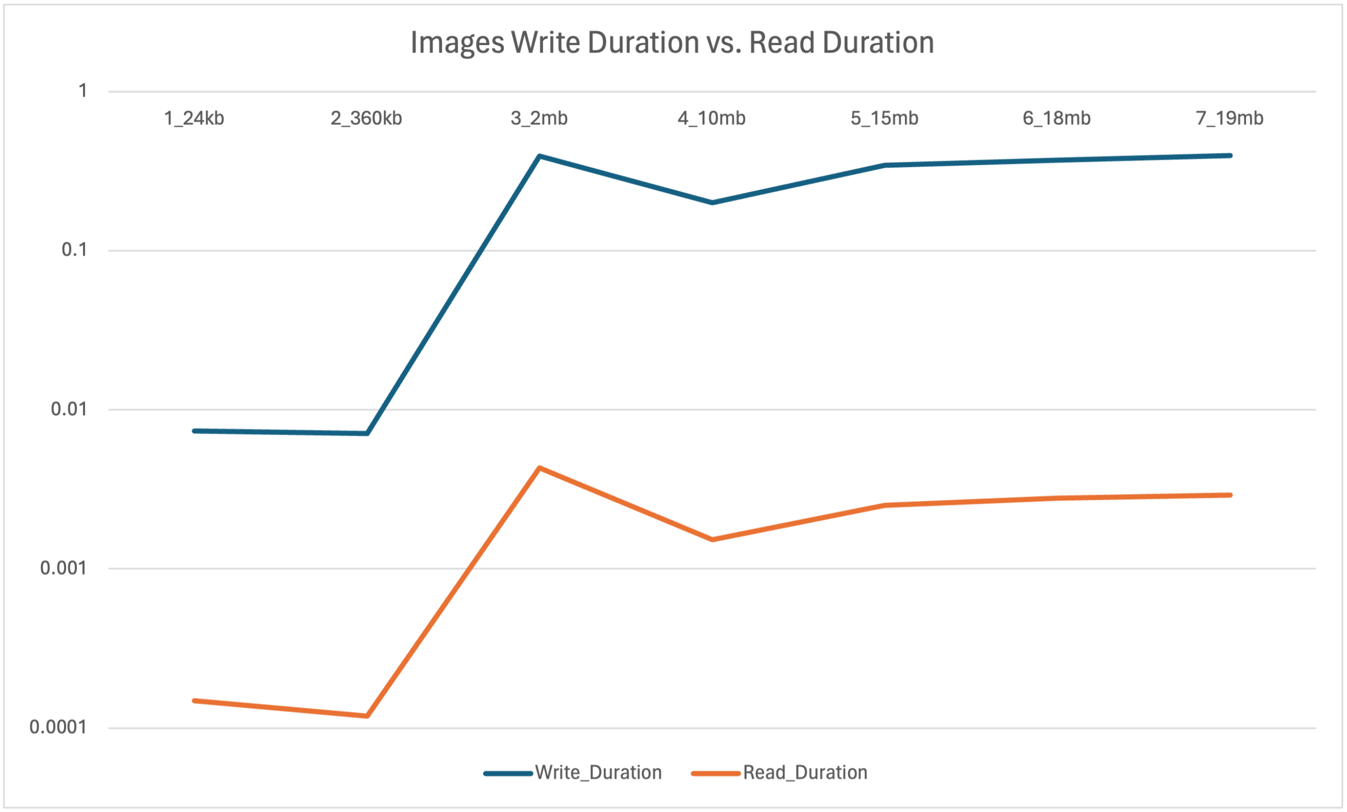
```
Set [s], Get [g], Clear Screen[c] or Quit [q]:
```

```
Set [s], Get [g], Clear Screen[c] or Quit [q]:
```

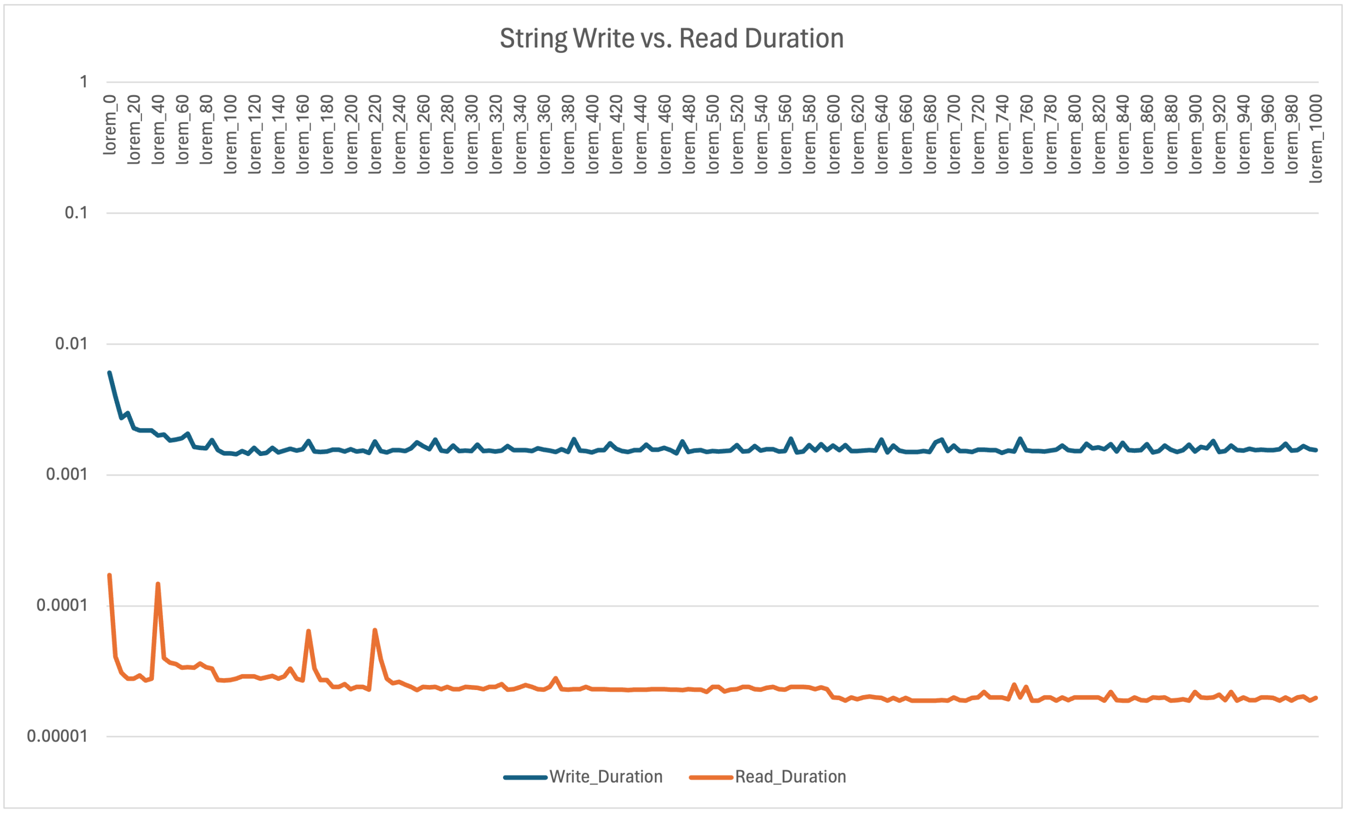
```
Set [s], Get [g], Clear Screen[c] or Quit [q]: s
Enter type <key,value> [kv], image [img], defined img [ai] lorem mode[l]: ai
File 1_24kb uploaded
File 2_360kb uploaded
File 3_2mb uploaded
File 4_10mb uploaded
File 5_15mb uploaded
File 6_18mb uploaded
File 7_19mb uploaded
Set [s], Get [g], Clear Screen[c] or Quit [q]: s
Enter type <key,value> [kv], image [img], defined img [ai] lorem mode[l]: l
Set 1000 lorem ipsum key, value pairs.
Set [s], Get [g], Clear Screen[c] or Quit [q]:
```

Measurements

- Graph 1: Write and Read Images Duration:



- Graph 2: Duration of set and get 200 lorem ipsum, each string increase by 5 words

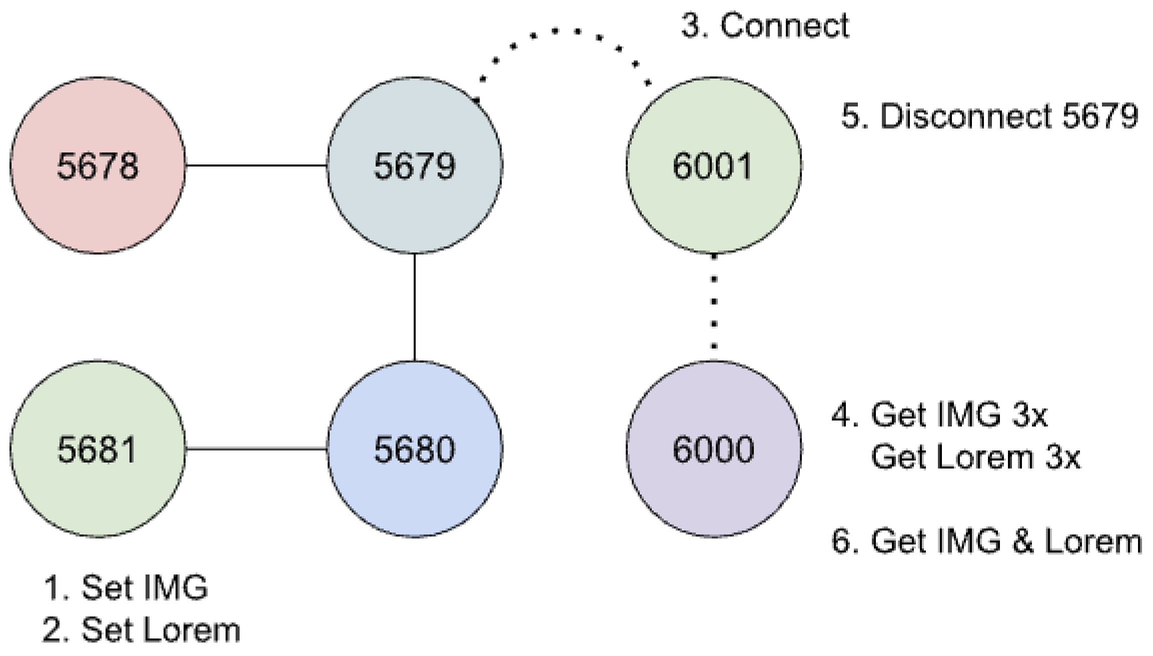


Discussion

Purpose of experience 2 is to measure the performance of write and read if a node is far away from inital node.

- In Graph 1, Write duration is significantly slower than read duration (Images).
  - Write an image to DHT system is about `243 ms` slower than read an image.
- In Graph 2, Write duration is also significantly slower than read duration (Strings).
  - Write a string to DHT system is about `1.62 ms` slower than read a string.

Experience 3

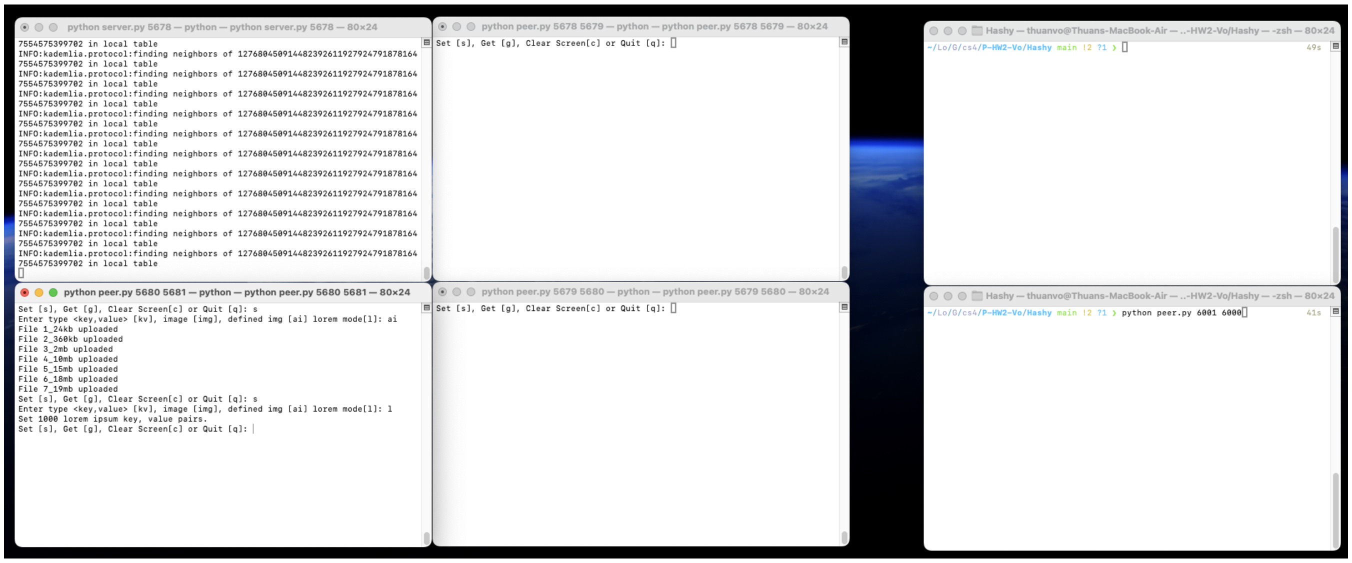


Steps

1. Start Node `5678`
2. Start Node `5679` and connect to `5678`
3. Start Node `5680` and connect to `5679`
4. Start Node `5681` and connect to `5680`
5. Set Images and Lorem at Node `5681`

6. Start Node `6001` and connect to Node `5679`
7. Start Node `6000` and connect to Node `6001`
8. Get Images and Lorem at Node `6000` **3 times** to measure
9. Disconnect Node `5679` from the Network
10. Get Images and Lorem at Node `6000`

Write before `6000` and `6001` connect:



```
python server.py 5678 — python — python server.py 5678 — 80×24
7554575399702 in local table
INFO:kademlia.protocol:finding neighbors of 127680450914482392611927924791878164
7554575399702 in local table
INFO:kademlia.protocol:finding neighbors of 127680450914482392611927924791878164
7554575399702 in local table
INFO:kademlia.protocol:finding neighbors of 127680450914482392611927924791878164
7554575399702 in local table
INFO:kademlia.protocol:finding neighbors of 127680450914482392611927924791878164
7554575399702 in local table
INFO:kademlia.protocol:finding neighbors of 127680450914482392611927924791878164
7554575399702 in local table
INFO:kademlia.protocol:finding neighbors of 127680450914482392611927924791878164
7554575399702 in local table
INFO:kademlia.protocol:finding neighbors of 127680450914482392611927924791878164
7554575399702 in local table
INFO:kademlia.protocol:finding neighbors of 127680450914482392611927924791878164
7554575399702 in local table
INFO:kademlia.protocol:finding neighbors of 127680450914482392611927924791878164
7554575399702 in local table
INFO:kademlia.protocol:finding neighbors of 127680450914482392611927924791878164
7554575399702 in local table
INFO:kademlia.protocol:finding neighbors of 127680450914482392611927924791878164
7554575399702 in local table
▯
```

```
python peer.py 5678 5679 — python — python peer.py 5678 5679 — 80×24
Set [s], Get [g], Clear Screen[c] or Quit [q]: ▯
```

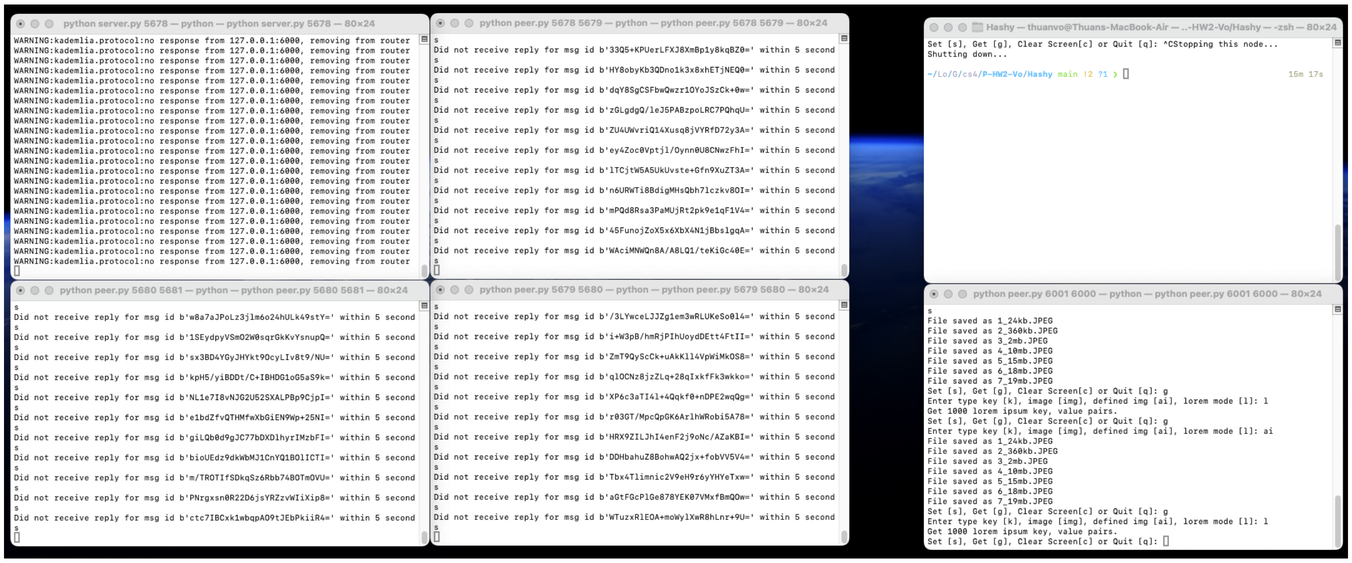
```
Hashy — thuanvo@Thuans-MacBook-Air — ..-HW2-Vo/Hashy — -zsh — 80×24
~/Lo/G/cs4/P-HW2-Vo/Hashy main !2 ?1 › ▯                                   49s
```

```
python peer.py 5680 5681 — python — python peer.py 5680 5681 — 80×24
Set [s], Get [g], Clear Screen[c] or Quit [q]: s
Enter type <key,value> [kv], image [img], defined img [ai] lorem mode[l]: ai
File 1_24kb uploaded
File 2_360kb uploaded
File 3_2mb uploaded
File 4_10mb uploaded
File 5_15mb uploaded
File 6_18mb uploaded
File 7_19mb uploaded
Set [s], Get [g], Clear Screen[c] or Quit [q]: s
Enter type <key,value> [kv], image [img], defined img [ai] lorem mode[l]: l
Set 1000 lorem ipsum key, value pairs.
Set [s], Get [g], Clear Screen[c] or Quit [q]: |
```

```
python peer.py 5679 5680 — python — python peer.py 5679 5680 — 80×24
Set [s], Get [g], Clear Screen[c] or Quit [q]: ▯
```

```
Hashy — thuanvo@Thuans-MacBook-Air — ..-HW2-Vo/Hashy — -zsh — 80×24
~/Lo/G/cs4/P-HW2-Vo/Hashy main !2 ?1 › python peer.py 6001 6000▯            41s
```

Write `3x time`



```
python server.py 5678 — python — python server.py 5678 — 80×24
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
WARNING:kademlia.protocol:no response from 127.0.0.1:6000, removing from router
▯
```

```
python peer.py 5678 5679 — python — python peer.py 5678 5679 — 80×24
s
Did not receive reply for msg id b'33Q5+KPUerLFXJ8XmBp1y8kqBZ0=' within 5 second
s
Did not receive reply for msg id b'HY8obyKb3QDno1k3x8xhETjNEQ0=' within 5 second
s
Did not receive reply for msg id b'dqY8SgCSFbwQwzr1OYoJSzCk+0w=' within 5 second
s
Did not receive reply for msg id b'zGLgdgQ/leJ5PABzpoLRC7PQhqU=' within 5 second
s
Did not receive reply for msg id b'ZU4UWvriQ14Xusq8jVYRfD72y3A=' within 5 second
s
Did not receive reply for msg id b'ey4Zoc0Vptjl/Oynn0U8CNwzFhI=' within 5 second
s
Did not receive reply for msg id b'lTCjtW5A5UkUvste+Gfn9XuZT3A=' within 5 second
s
Did not receive reply for msg id b'n6URWTi8BdigMHsQbh7lczkv8OI=' within 5 second
s
Did not receive reply for msg id b'mPQd8Rsa3PaMUjRt2pk9e1qF1V4=' within 5 second
s
Did not receive reply for msg id b'45FunojZoX5x6XbX4N1jBbslgqA=' within 5 second
s
Did not receive reply for msg id b'WAciMNWQn8A/A8LQ1/teKiGc40E=' within 5 second
s
▯
```

```
Hashy — thuanvo@Thuans-MacBook-Air — ..-HW2-Vo/Hashy — -zsh — 80×24
Set [s], Get [g], Clear Screen[c] or Quit [q]: ^CStopping this node...
Shutting down...

~/Lo/G/cs4/P-HW2-Vo/Hashy main !2 ?1 › ▯                               15m 17s
```

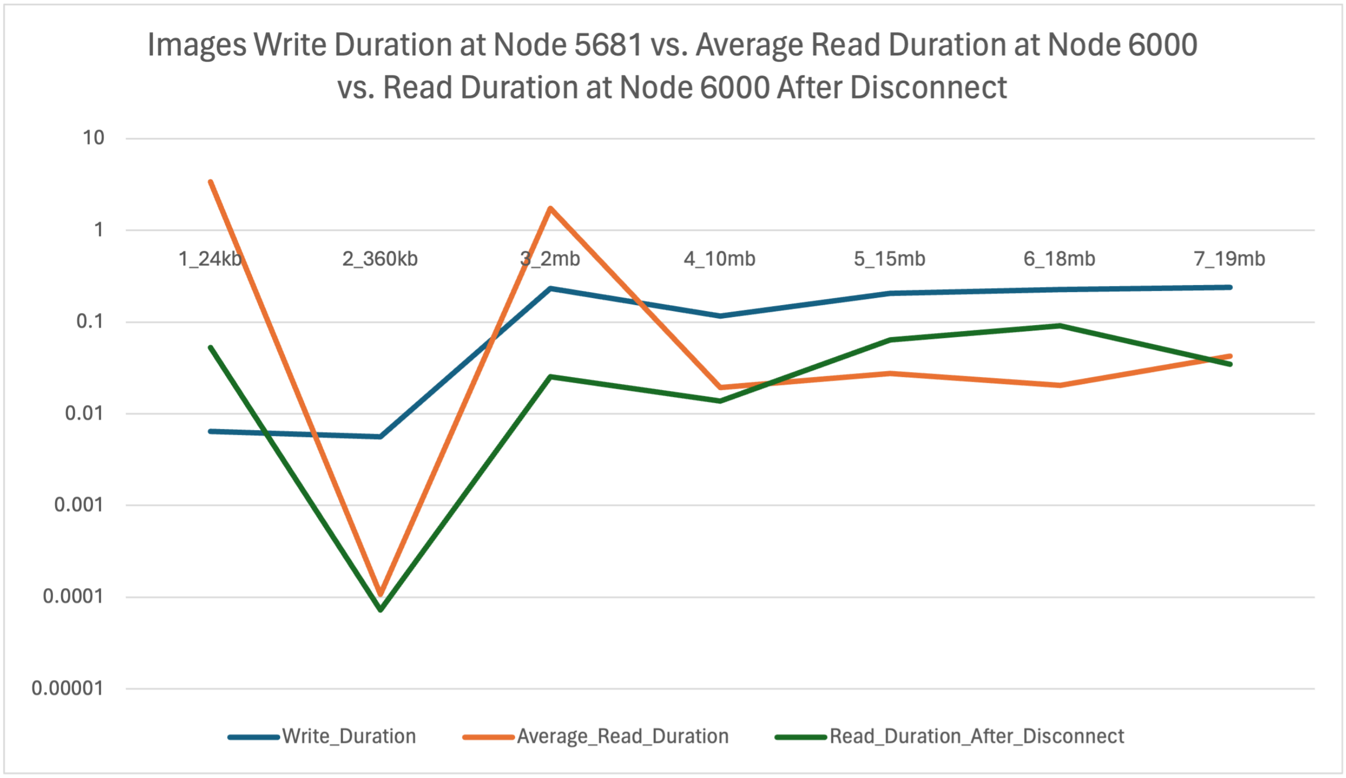
```
python peer.py 5680 5681 — python — python peer.py 5680 5681 — 80×24
s
Did not receive reply for msg id b'w8a7aJPoLz3jlm6o24hULk49stY=' within 5 second
s
Did not receive reply for msg id b'1SEydpyVSmO2W0sqrGkKvYsnupQ=' within 5 second
s
Did not receive reply for msg id b'sx3BD4YGyJHYkt9OcyLIv8t9/NU=' within 5 second
s
Did not receive reply for msg id b'kpH5/yiBDDt/C+IBHDG1oG5aS9k=' within 5 second
s
Did not receive reply for msg id b'NL1e7I8vNJG2U52SXALPBp9CjpI=' within 5 second
s
Did not receive reply for msg id b'e1bdZfvQTHMfwXbGiEN9Wp+25NI=' within 5 second
s
Did not receive reply for msg id b'giLQb0d9gJC77bDXDlhyrIMzbFI=' within 5 second
s
Did not receive reply for msg id b'bioUEdz9dkWbMJ1CnYQ1BOlICTI=' within 5 second
s
Did not receive reply for msg id b'm/TROTIfSDkqSz6Rbb74BOTmOVU=' within 5 second
s
Did not receive reply for msg id b'PNrgxsn0R22D6jsYRZzvWIiXip8=' within 5 second
s
Did not receive reply for msg id b'ctc7IBCxk1wbqpAO9tJEbPkiiR4=' within 5 second
s
▯
```

```
python peer.py 5679 5680 — python — python peer.py 5679 5680 — 80×24
s
Did not receive reply for msg id b'/3LYwceLJJZg1em3wRLUKeSo0l4=' within 5 second
s
Did not receive reply for msg id b'i+W3pB/hmRjPIhUoydDEtt4FtII=' within 5 second
s
Did not receive reply for msg id b'ZmT9QyScCk+uAkKll4VpWiMkOS8=' within 5 second
s
Did not receive reply for msg id b'qlOCNz8jzZLq+28qIxkfFk3wkko=' within 5 second
s
Did not receive reply for msg id b'XP6c3aTI4l+4Qqkf0+nDPE2wqQg=' within 5 second
s
Did not receive reply for msg id b'r03GT/MpcQpGK6ArlhWRobi5A78=' within 5 second
s
Did not receive reply for msg id b'HRX9ZILJhI4enF2j9oNc/AZaKBI=' within 5 second
s
Did not receive reply for msg id b'DDHbahuZ8BohwAQ2jx+fobVV5V4=' within 5 second
s
Did not receive reply for msg id b'Tbx4Tlimnic2V9eH9r6yYHYeTxw=' within 5 second
s
Did not receive reply for msg id b'aGtFGcPlGe878YEK07VMxfBmQOw=' within 5 second
s
Did not receive reply for msg id b'WTuzxRlEOA+moWylXwR8hLnr+9U=' within 5 second
s
▯
```

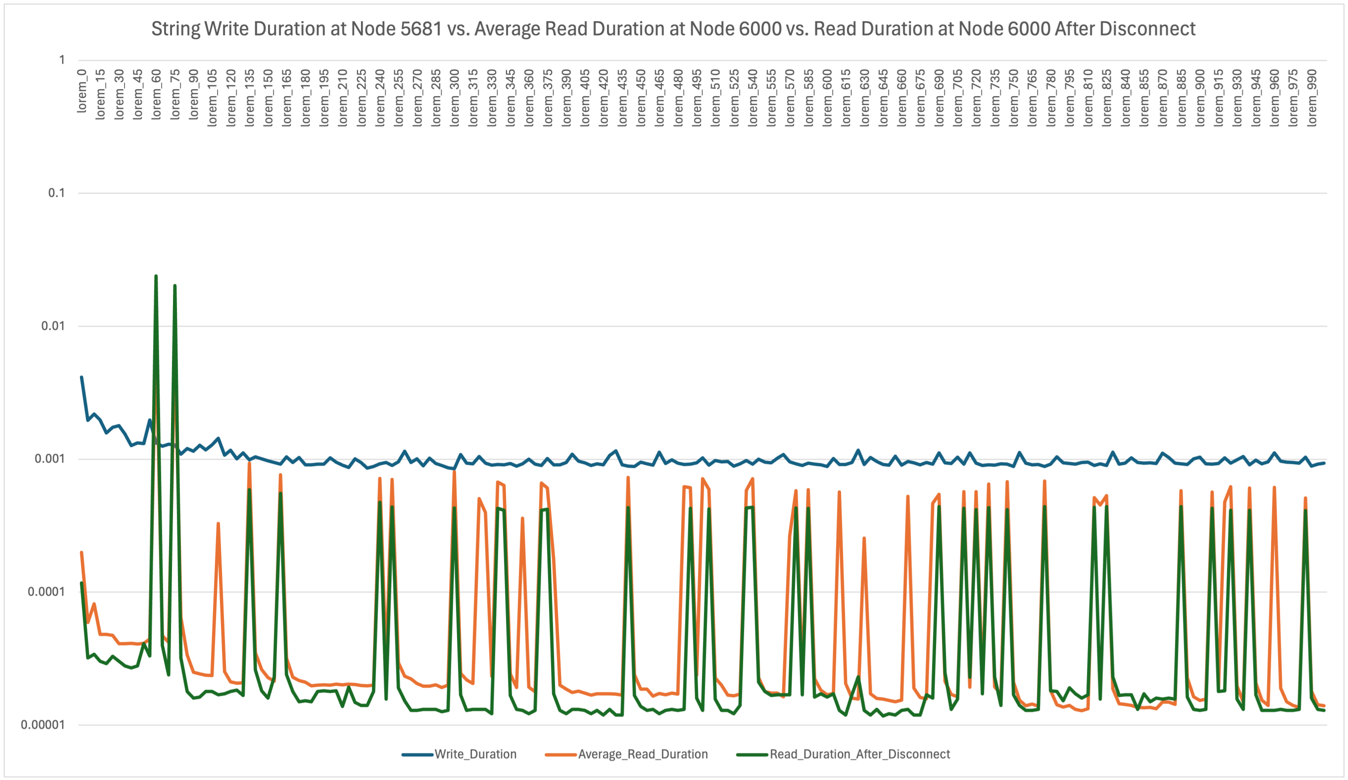
```
python peer.py 6001 6000 — python — python peer.py 6001 6000 — 80×24
s
File saved as 1_24kb.JPEG
File saved as 2_360kb.JPEG
File saved as 3_2mb.JPEG
File saved as 4_10mb.JPEG
File saved as 5_15mb.JPEG
File saved as 6_18mb.JPEG
File saved as 7_19mb.JPEG
Set [s], Get [g], Clear Screen[c] or Quit [q]: g
Enter type key [k], image [img], defined img [ai], lorem mode [l]: l
Get 1000 lorem ipsum key, value pairs.
Set [s], Get [g], Clear Screen[c] or Quit [q]: g
Enter type key [k], image [img], defined img [ai], lorem mode [l]: ai
File saved as 1_24kb.JPEG
File saved as 2_360kb.JPEG
File saved as 3_2mb.JPEG
File saved as 4_10mb.JPEG
File saved as 5_15mb.JPEG
File saved as 6_18mb.JPEG
File saved as 7_19mb.JPEG
Set [s], Get [g], Clear Screen[c] or Quit [q]: g
Enter type key [k], image [img], defined img [ai], lorem mode [l]: l
Get 1000 lorem ipsum key, value pairs.
Set [s], Get [g], Clear Screen[c] or Quit [q]: ▯
```

Measurements:

- Graph 1: Images Write Duration at Node 5681 vs. Average Read Duration at Node 6000 after connect vs. Read Duration at Node 6000 after disconnect Node 6001 from the network:



- Graph 2: String Write Duration at Node 5681 vs. Average Read Duration at Node 6000 after connect vs. Read Duration at Node 6000 after disconnect Node 6001 from the network:



Discussion

The purpose of experience 3 is to measure performance of get data from DHT system if a node joined the network after the data is set. To see if the new node takes more time to discover the data in the existing network. Also, we want to see if a neighbor peer left the network, does it affect the ability to discover the network of its peers.

As we can observe from graph 1, the data has been set, after Node `6000` joins the network:

- It take Node `6000` slower than Experience 1 and 2 to read, Average of `746 ms`.
- After 6001 left the network, the node 6000 lost a peer, it takes 6000 about `40 ms` to read from the network.

As we can observe from graph 2:

- It takes Node `6000` about average of `0.23 ms` to read strings from the network after joining.
- After 6001 left the network, it takes Node `6000` about `0.30 ms` to read strings from the networks.

General Discussion

> **Each experience above has its own discussion.**

Compare 3 experiences:

Read/Write	Experience 1	Experience 2
Avg Write Img	223.0 ms	245.5 ms
Avg Read Img	2.11 ms	2.04 ms
Avg Write Str	1.47 ms	1.65 ms
Avg Read Str	0.03 ms	0.25 ms

- Experience 3:
  - Images:
    - Average write duration: `147.37 ms`
    - Average read duration after join: `746.4 ms`
    - Average read duration after disconnect peer: `40.2 ms`
  - Strings:
    - Average write duration: `1.03 ms`
    - Average read duration after join: `0.23 ms`
    - Average read duration after disconnect peer: `0.30 ms`

From the data collected above:

- The more nodes in the network, the further the set/get node from the initial node, it may take more time to write to the network. However, the performance is not significantly different for writing/setting to the network.
- If the node joins after the data is set, it may take the node more time to discover and get data from the existing network.
  - Experience 1 and 2: it takes about `2 ms` to read/get images.
  - However, in experience 3, after node 6000 joins the network, it takes `node 6000` up to `750 ms` to explore and get the images.
  - Why does experience 3 take less time to write data (images/strings) to the network?
    - It could be because there are fewer nodes than in experiences 1 and 2 when it writes the data in step 5.

- If the node joins lost a peer it connected to, it also takes more time to read data from the network.
  - The average time to read images (experience 3) is `40.2 ms`, slower than both experiences 1 and 2 (`2 ms`).
  - The average time to read strings in experience 3 is `0.30 ms`, also slower than experience 1 (`0.03 ms`), but not significantly compared to experience 2 (`0.25 ms`).

Some observations from experiences 1, 2, 3:

- The heavier the images, the longer it takes to write and read from the networks.
- Longer strings do not significantly take longer to write to the networks.

Why do I break images into chunks of bytes?

- For each key:value, the maximum length of `value` is 8000. Therefore, the only way to save an image in the DHT system is to save many chunks of bytes.
- Each image can have hundreds of chunks, each chunk is full of 8000 bytes. While each regular string doesn't reach the maximum of 8000 bytes. Therefore, getting an image of 100 chunks takes significantly more time and is slower than getting 100 key:value pairs.

How do I break them into chunks?

```
For any image, lower the quality to 85% of the original quality.
Then, it will be broken into chunks of 8000 bytes.

To save in the DHT networks:

1. Set IMAGE_NAME:NUMBER_OF_CHUNKS
2. Set all the chunks with the key IMAGE_NAME_i where i is the index of the
chunk.
```

- The DHT system is fast in the 3 experiences above, and it can have many applications. However, consider encrypting data before sending it to the system to securely store data since everyone in the network can access your data by having the key.

References

- Distributed Hash Tables with Kademlia
- Kademlia Documentation
- Pillow Documentation
- Async IO in Python: A Complete Walkthrough