

Computer Communication

1. Tests

1.1. IP and Port Assignments

I did not have multiple computers at my disposal so I used multiple terminals instead and had all services running on localhost

Bitsource listens on: 127.0.0.1 6000 (recv from client) and 127.0.0.1 6001 (publish data)

Pythagoras listens on: 127.0.0.1 6002

Tally listens on: 127.0.0.1 6003

Client listens on: 127.0.0.1 6004

Always_yes and judge do not bind any to any url

1.2. Example Output

1.2.1 All Services

```
3.2 *
* *****
3 * *****
*****
***
2.8 -----
0 500 1000 1500 2000 2500 3000 3500
|
|
|
3.8 |
|
3.6 |
|
3.4 |
|
3.2 |
* *****
* *****
3 * *****
*****
***
2.8 -----
0 500 1000 1500 2000 2500 3000 3500

^CTraceback (most recent call last):
  File "/home/marius/projects/University/exchange/cc/assign_1/main.py", line 180, in <module>
    main(zmq.Context(), arguments)
  File "/home/marius/projects/University/exchange/cc/assign_1/main.py", line 59, in main
    time.sleep(150)
KeyboardInterrupt

(assign_1) + assign_1 git:(main) X python3 main.py tally -u 127.0.0.1 -p 6703 -cu 127.0.0.1 -c
p 6704
Starting Tally on: tcp://127.0.0.1:6703
Starting Tally
(assign_1) + assign_1 git:(main) X

Starting pythagoras
^CTraceback (most recent call last):
  File "/home/marius/projects/University/exchange/cc/assign_1/main.py", line 180, in <module>
    main(zmq.Context(), arguments)
  File "/home/marius/projects/University/exchange/cc/assign_1/main.py", line 59, in main
    time.sleep(150)
KeyboardInterrupt

(assign_1) + assign_1 git:(main) X python3 main.py pythagoras -u 127.0.0.1 -p 6702
Starting the pythagoras service on: tcp://127.0.0.1:6702
Starting pythagoras
(assign_1) + assign_1 git:(main) X

[0] 1:nvln- 2:python3*

[8/48042] sending: 11101111001110010111001000011110010010010101001111100110100
sending: 111000011000001011010001100100010101011100110100100011111111
sending: 11000011001001001101111001010110001000000100001000011000100001
sending: 0101001011111000010000111111010101000010000111111010100101011
sending: 00001101110111100101110101011101011101101000010001110000101
sending: 1011011100111001110000010100100001000000001001101010001110000
sending: 11011101000011110001101001000111000100110101010010110000101000
sending: 0101001110110001001100100010011011000000001100110010010100101
sending: 1000101000111001010000001110001110011111110000010111101001100
sending: 1101101010110101011101101101010000010100111000101001110111100
sending: 11011101010001100100000111110100111001100110101010111001011
sending: 11110101001111110111110110000101110110000100111000101100000101
sending: 0101000100110010011101010101001001000001011001110000101011010
sending: 110100100111110101001110101010000110100100000101100000010010
sending: 01000111111001011100011011000100110101010001000010100110001
sending: 0000110100101101100111010000100100100100100011000110001100100
sending: 010100001111011111100000001001100000010111111000101010100001
sending: 00110000011001101111011000110110110000101101010101010011100
^CTraceback (most recent call last):
  File "/home/marius/projects/University/exchange/cc/assign_1/main.py", line 180, in <module>
    main(zmq.Context(), arguments)
  File "/home/marius/projects/University/exchange/cc/assign_1/main.py", line 59, in main
    time.sleep(150)
KeyboardInterrupt

(assign_1) + assign_1 git:(main) X

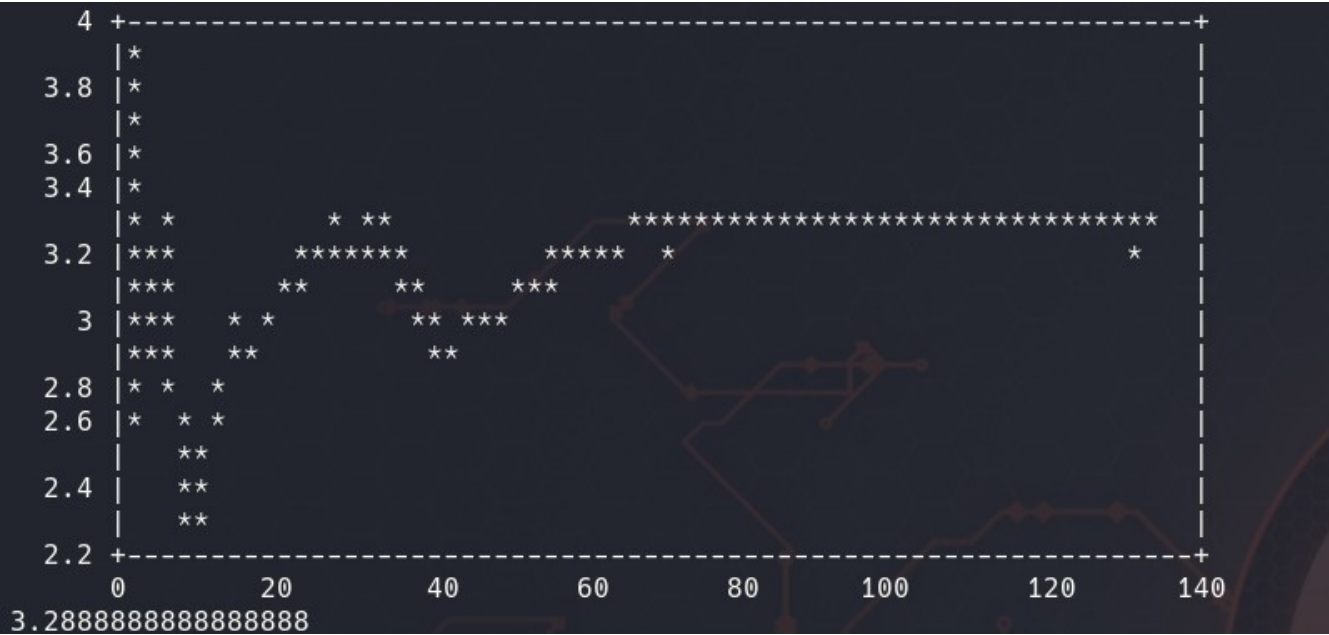
Received: 0111010100111011000110110101111000011101110001100110000000000
Sending No
Received: 111101101101100111001101010101001010011111010010100110010011
Sending No
Received: 01001010010100010011110111000000010011011100111010100000101
Sending Yes
Received: 111111111011000111101000011101000011101110110100100100100010
Sending No
Received: 11100010010001111001100110011011011000001000010101010011
Sending Yes
Received: 0111010001010101110101001001111111100100001000011110101011
(assign_1) + assign_1 git:(main) X

File "/home/marius/projects/University/exchange/cc/assign_1/main.py", line 59, in main
time.sleep(150)
KeyboardInterrupt

(assign_1) + assign_1 git:(main) X k
zsh: k: command not found...
(assign_1) + assign_1 git:(main) X python3 main.py always_yes -u 127.0.0.1 -p 6701 -d 127.0.0.1 -dp 6703
Always_yes service receiving data from: tcp://127.0.0.1:6701
Sending data to tcp://127.0.0.1:6703
starting always_yes
(assign_1) + assign_1 git:(main) X

"localhost.localdomain" 16:59 19-May-21
```

1.2.2 Client Only



Here I stopped the run early to show that at the start there were values at above 3.8 below 2.4, but then the calculation gets closer to the real pi value

2. Dependencies and how to install them

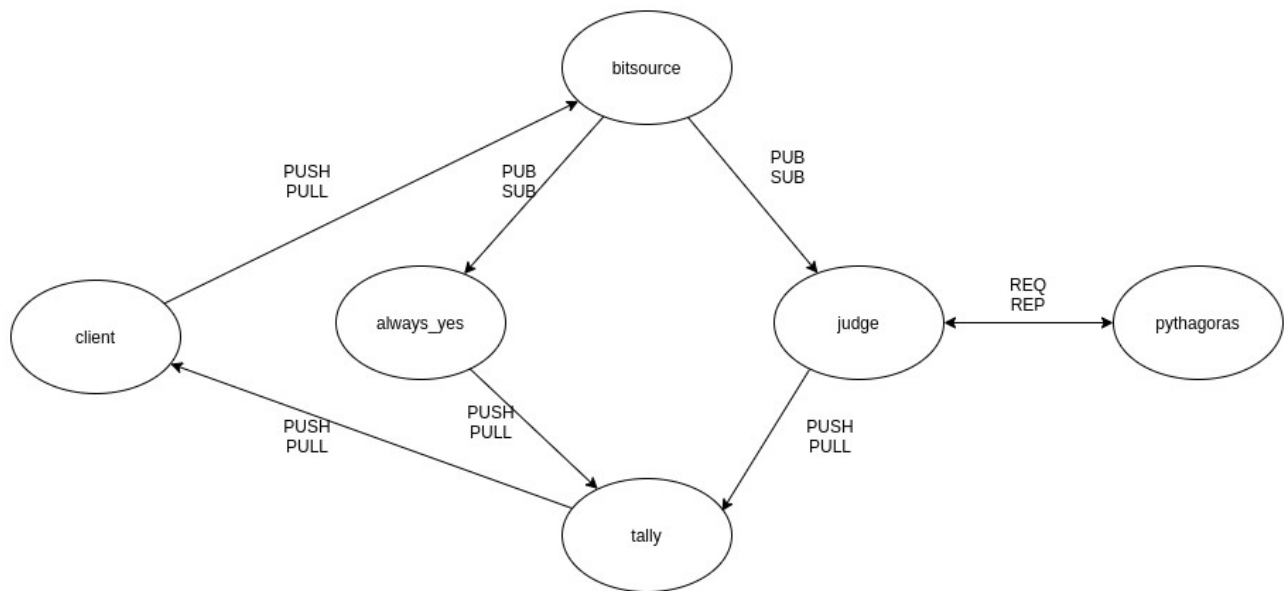
The project depends on pyzmq and termplotlib. To manage these dependencies I used pipenv (<https://pypi.org/project/pipenv/>). Once installed the virtual environmental can be activated with “pipenv shell” and the dependencies can then be installed with “pipenv install”.

3. Process View

The following screenshot shows the output of the command “ss -tulpn”

tcp	LISTEN	0	100	127.0.0.1:6700	0.0.0.0:*	users:(("python3",pid=79435,fd=10))
tcp	LISTEN	0	100	127.0.0.1:6701	0.0.0.0:*	users:(("python3",pid=79435,fd=13))
tcp	LISTEN	0	100	127.0.0.1:6702	0.0.0.0:*	users:(("python3",pid=79462,fd=10))
tcp	LISTEN	0	100	127.0.0.1:6703	0.0.0.0:*	users:(("python3",pid=79469,fd=10))
tcp	LISTEN	0	100	127.0.0.1:6704	0.0.0.0:*	users:(("python3",pid=79428,fd=11))

4. Service Architecture



I choose PUSH and PULL sockets because PUBSUB doesn't seem reasonable since we are not communicating with multiple other services at the same time and REQ/REP doesn't seem reasonable because we do not need to respond to tally neither does the bitsource need to response to us. So PUSH PULL was the only method that I couldn't find an argument against

5. Running the services

You can get a separate help menu for each service using:

```
python3 main.py client -h
python3 main.py bitsource -h
```

and so on

Here are the exact commands I used for my test case shown on the first page

```
python3 main.py client -u 127.0.0.1 -p 6704 -bu 127.0.0.1 -bp 6700
python3 main.py bitsource -u 127.0.0.1 -p 6701 -lcu 127.0.0.1 -lcp 6700
python3 main.py judge -bu 127.0.0.1 -bp 6701 -pu 127.0.0.1 -pp 6702 -tu 127.0.0.1 -tp 6703
python3 main.py always_yes -u 127.0.0.1 -p 6701 -d 127.0.0.1 -dp 6703
python3 main.py tally -u 127.0.0.1 -p 6703 -cu 127.0.0.1 -cp 6704
python3 main.py pythagoras -u 127.0.0.1 -p 6702
```