

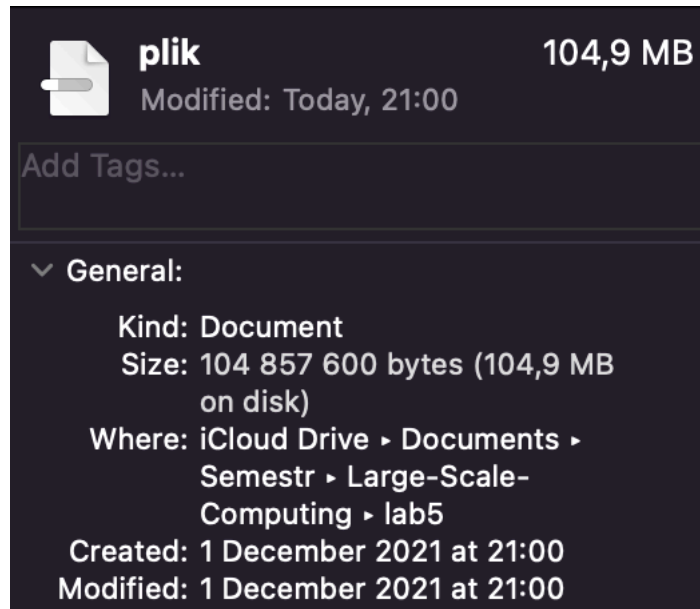
Large Scale Computing

Lab 5

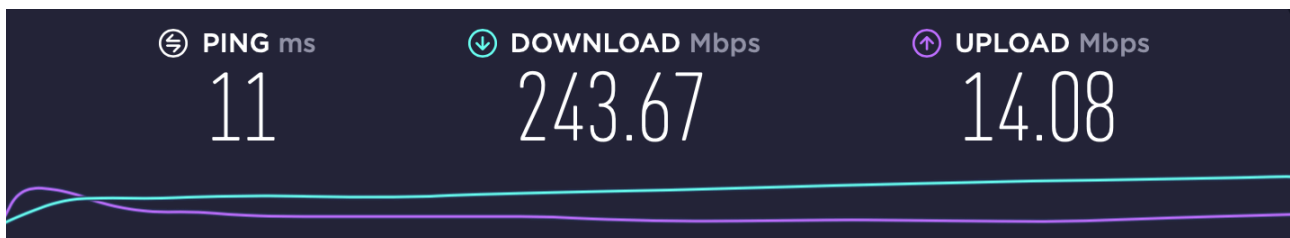
Oscar Teeninga

1. Plik

Pobrałem plik testowy z <http://noc.pirx.pl>, wybierając ten o wielkości 100MB.



2. Prędkość mojego internetu



3. Pomiar czasu dla AWS S3 CLI

```
[oscarteeninga@0scars-MacBook-Pro-16 lab5 % aws s3 cp plik s3://bucket-oscarteeninga
upload: ./plik to s3://bucket-oscarteeninga/plik
[oscarteeninga@0scars-MacBook-Pro-16 lab5 % aws s3 cp s3://bucket-oscarteeninga/plik plik_download_cli
download: s3://bucket-oscarteeninga/plik to ./plik_download_cli
```

Objects (1)
Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Copy S3 URI

Copy URL

Download

Open

Delete

Actions ▼

Create folder

Upload

☐

Name ▲

Type ▼

Last modified ▼

Size ▼

Storage class ▼

☐

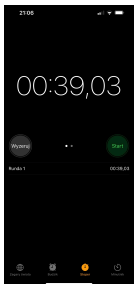
plik

-

December 1, 2021, 21:05:54 (UTC+01:00)

100.0 MB

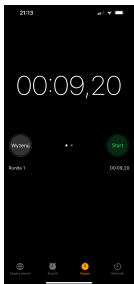
Standard



Pomiary wykonywałem telefonem

Zgodnie z telefonem, czas upload to było 39 sekund. Dane tutaj pokrywają się z maksymalną przepustowością samego internetu, zatem można spodziewać się, że wielkość ta może być nawet większa dla S3.

Upload: $100\text{MB} / 39\text{s} = 2,56\text{ MB/s} = 20,5\text{ Mb/s}$



Zgodnie z telefonem, czas download to było 9,2 sekund. Tutaj wynik jest niższy niż maksymalna przepustowość, natomiast jest to wynik i tak zadowalający.

Download: $100\text{ MB} / 9,2\text{s} = 10,96\text{ MB/s} = 87\text{ Ms/s}$

4. Pomiar czasu dla programu na lokalnym komputerze

Program zaimplementowałem w Pythonie, przy pomocy biblioteki boto.

```
1 import time
2 import boto3
3 import botocore
4
5 file_name = 'plik'
6 bucket_name = 'bucket-oscarteeninga'
7
8 start_time = time.time()
9
10 s3 = boto3.resource('s3')
11
12 file = s3.Object(bucket_name, file_name)
13
14 result = file.put(Body=open(file_name, 'rb'))
15
16 response = result.get('ResponseMetadata')
17
18 print("Upload time: " + str(time.time() - start_time))
19
20 start_time = time.time()
21 if response.get('HTTPStatusCode') != 200:
22     print("File not upload")
23 else:
24     print("File uploaded")
25
26 try:
27     s3.Bucket(bucket_name).download_file(file_name, file_name)
28 except e:
29     print(e)
30
31 print("Download time: " + str(time.time() - start_time))
```

```
[oscarteeninga@Oscars-MacBook-Pro-16 lab5 % python3 up_down.py
Upload time: 66.39594602584839
File uploaded
Download time: 21.461797952651978
```

Wyniki są następujące, zgodnie z poprzednim wzorem:

Upload: $100\text{MB} / 66,4\text{ s} = 1,5\text{ MB/s} = 12,1\text{ Mb/s}$

Download: $100\text{ MB} / 21,5\text{ s} = 4,6\text{ MB/s} = 37,2\text{ Mb/s}$

5. Pomiar czasu dla programu na Amazon EC2

Korzystając z scp przekopiowałem plik na instancję EC2.

```
oscarteeninga@Oscars-MacBook-Pro-16 lab5 % scp -i aws-oscarteeninga-ec2.pem plik ubuntu@ec2-3-89-250-35.compute-1.amazonaws.com:~/.  
plik 100% 100MB 3.2MB/s 00:31
```

Osiągnięta prędkość upload to 3,2 MB/s

```
[ubuntu@ip-172-31-91-148:~]$ python3 up_down.py  
Upload time: 1.3235759735107422  
File uploaded  
Download time: 2.0954208374023438
```

Przy uruchomieniu programu na maszynie EC2 osiągnięte prędkości są imponujące względem tych, które osiągnąłem lokalnie.

Upload: 100 MB / 1,32 s = 75 MB/s = 600 Mb/s

Download: 100 MB / 2,1 s = 47,6 MB/s = 381 Mb/s