Timing_study

August 20, 2020

0.1 Assignment: Timing Study

Team_AG : Akshay Bankar(2019201011) and Gajanan Modi (2019201049)

0.1.1 Aim: To plot and compare access time and read-rate for commercial hard disk drives (HDD) and solid-state drives (SSD).

##Using Disks utility provided in Ubuntu:

- Ubuntu provides a disk performance utility for benchmarking. It lets us set sample size for varying the block transfer size and gives average performance over some fixed number of samples.
- We tested with,

  sample size = [1, 10, 100, 1000] with

number of samples = 1000

- We tested for following three disks (2 HDD and 1 SSD):

  A Seagate HDD, a transcend HDD and an SSD.

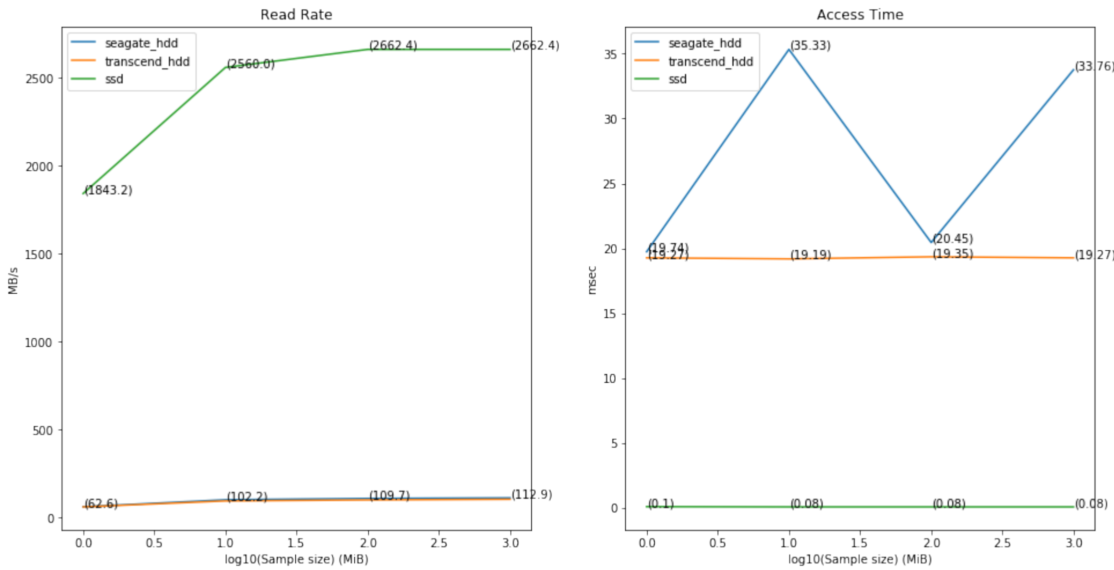
  The following plot shows their benchmarking results:

```
1. The first graph shows avg. read rate vs log10(Block size)
2. The second graph shows avg. access time vs log10(Block size).
```

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt

sample_sizes = [1,10,100,1000]
seagate_hdd = {'read_rate' : [62.6, 102.2, 109.7, 112.9], 'access_time' : [19.
↪74, 35.33, 20.45, 33.76]}
transcend_hdd = {'read_rate' : [61.7, 95.8, 101.4, 105.3], 'access_time' : [19.
↪27, 19.19, 19.35, 19.27]}
ssd = {'read_rate' : [1843.2, 2560.0, 2662.4, 2662.4], 'access_time' : [0.1, 0.
↪08, 0.08, 0.08]}
```

[ ]:
```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 8))
```

```
ax1.plot(np.log10(sample_sizes), seagate_hdd['read_rate'], label='seagate_hdd')␣
↪ # Plot some data on the axes.
for i_x, i_y in zip(np.log10(sample_sizes), seagate_hdd['read_rate']):
    ax1.text(i_x, i_y, '({})'.format(i_y))

ax1.plot(np.log10(sample_sizes), transcend_hdd['read_rate'],␣
↪label='transcend_hdd')  # Plot more data on the axes...
#for i_x, i_y in zip(np.log10(sample_sizes), transcend_hdd['read_rate']):
#    ax1.text(i_x, i_y, '({})'.format(i_y))

ax1.plot(np.log10(sample_sizes), ssd['read_rate'], label='ssd')  # Plot more␣
↪data on the axes...
for i_x, i_y in zip(np.log10(sample_sizes), ssd['read_rate']):
    ax1.text(i_x, i_y, '({})'.format(i_y))

ax2.plot(np.log10(sample_sizes), seagate_hdd['access_time'],␣
↪label='seagate_hdd')  # ... and some more.
for i_x, i_y in zip(np.log10(sample_sizes), seagate_hdd['access_time']):
    ax2.text(i_x, i_y, '({})'.format(i_y))

ax2.plot(np.log10(sample_sizes), transcend_hdd['access_time'],␣
↪label='transcend_hdd')  # ... and some more.
for i_x, i_y in zip(np.log10(sample_sizes), transcend_hdd['access_time']):
    ax2.text(i_x, i_y, '({})'.format(i_y))

ax2.plot(np.log10(sample_sizes), ssd['access_time'], label='ssd')  # ... and␣
↪some more.
for i_x, i_y in zip(np.log10(sample_sizes), ssd['access_time']):
    ax2.text(i_x, i_y, '({})'.format(i_y))

ax1.set_xlabel('log10(Sample size) (MiB)')  # Add an x-label to the axes.
ax1.set_ylabel('MB/s')  # Add a y-label to the axes.
ax2.set_xlabel('log10(Sample size) (MiB)')  # Add an x-label to the axes.
ax2.set_ylabel('msec')  # Add a y-label to the axes.
ax1.set_title("Read Rate")  # Add a title to the axes.
ax2.set_title("Access Time")  # Add a title to the axes.
ax1.legend()
ax2.legend()
fig.savefig('plot.jpg')
```



0.2 Using data from HDD and SSD manuals :

We picked data used for calculations from following links :

For Seagate HDD : https://www.disctech.com/Seagate-ST1000LM048-1TB-SATA-Hard-Drive

For Samsung (830 series) SSD : http://www.spkaa.com/wp-content/uploads/2013/01/SPK_HDTune.pdf

[2]:
```python
####--- Data for Seagate HDD ---####
track_per_sector=1/63
avg_seek_time=13/1000
avg_rotation=5.55/1000
time_to_cover_track=0.011
```

HDD access time = (avg_seek_time+avg_rotation)*(blocks)

[3]:
```python
####--- Calculate access time for HDD ---####
def hdd_access_time(blocks):
    return (avg_seek_time+avg_rotation)*(blocks)
```

HDD transfer time = (avg_seek_time+avg_rotation+time_to_cover_track)* track_per_sector * blocks

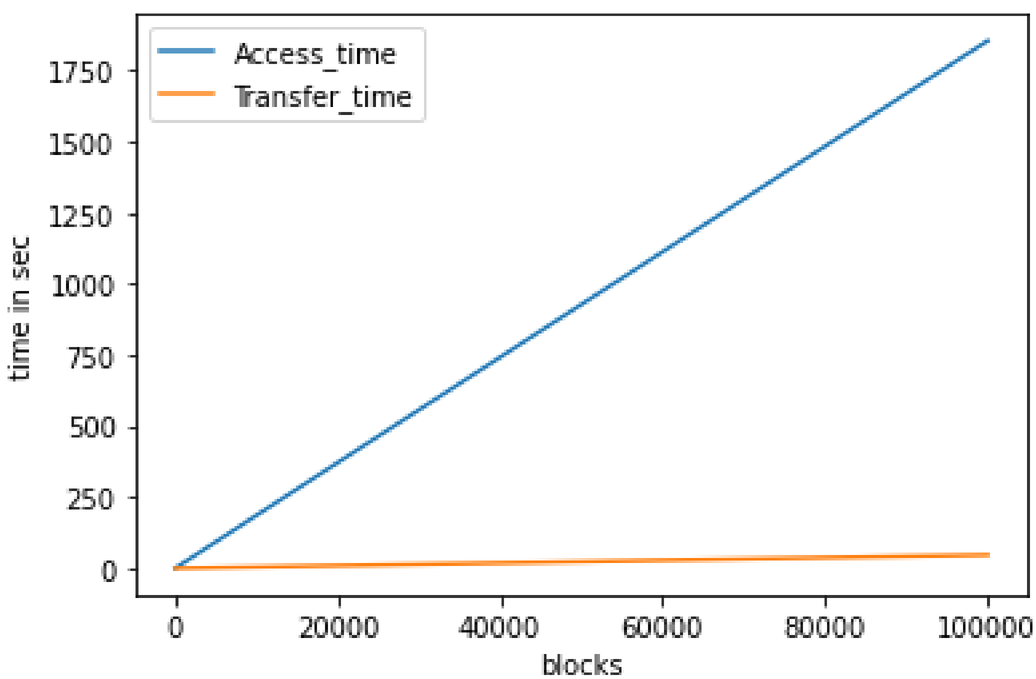
[4]:
```python
####--- Calculate tansfer time for HDD ---####
def hdd_transfer_time(blocks): #
    return (avg_seek_time+avg_rotation+time_to_cover_track)*track_per_sector*blocks
```

[25]:
```
timeForOne=1/46386.71
def HDDtoRAM(blocks):
    return timeForOne*blocks
```

[7]:
```
x=[1,10,100,1000,10000,100000]
y_access=[0]*len(x)
y_transfer=[0]*len(x)
for i in range(len(x)):
    y_access[i]=hdd_access_time(x[i])
    y_transfer[i]=hdd_transfer_time(x[i])
```

[9]:
```
import matplotlib.pyplot as plt

plt.plot(x,y_access)
plt.plot(x,y_transfer)
plt.legend(['Access_time','Transfer_time'])
plt.xlabel('blocks')
plt.ylabel('time in sec')
```

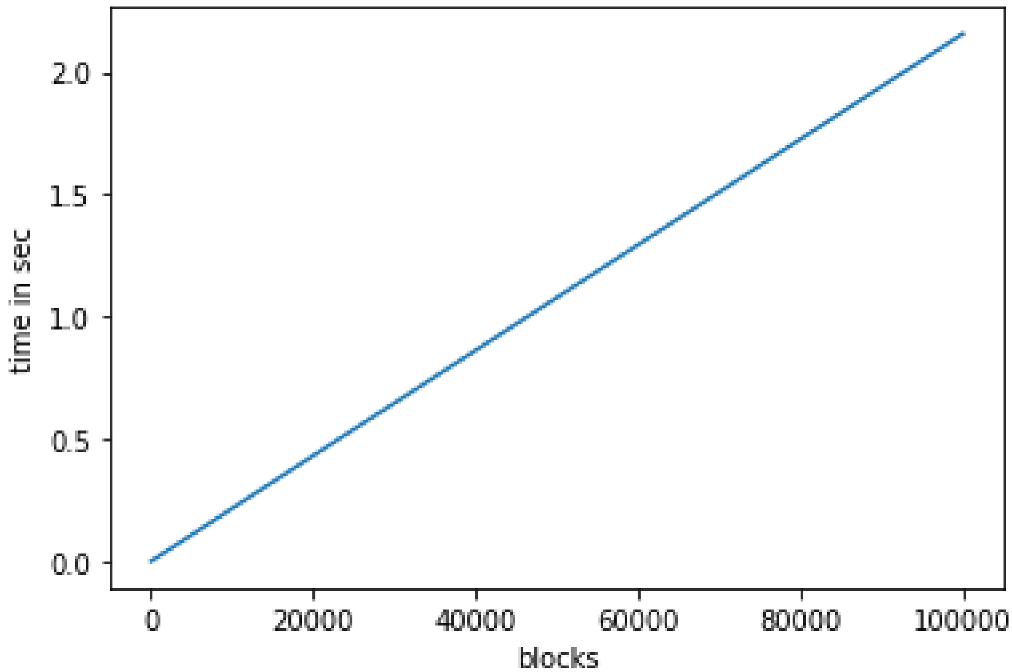
[9]: Text(0, 0.5, 'time in sec')



[26]:
```
ydatarate=[0]*len(x)
for i in range(len(x)):
    ydatarate[i]=HDDtoRAM(x[i])
plt.plot(x,ydatarate)
```

```python
plt.xlabel('blocks')
plt.ylabel('time in sec')
```

[26]: Text(0, 0.5, 'time in sec')



Calculations and plots for SSD (Samsung 830 series)

[30]:
```python
####--- Data for Samsung SSD ---####
consistent_latency = 0.2
bandwidth = 214.5
block_size = 1024
```

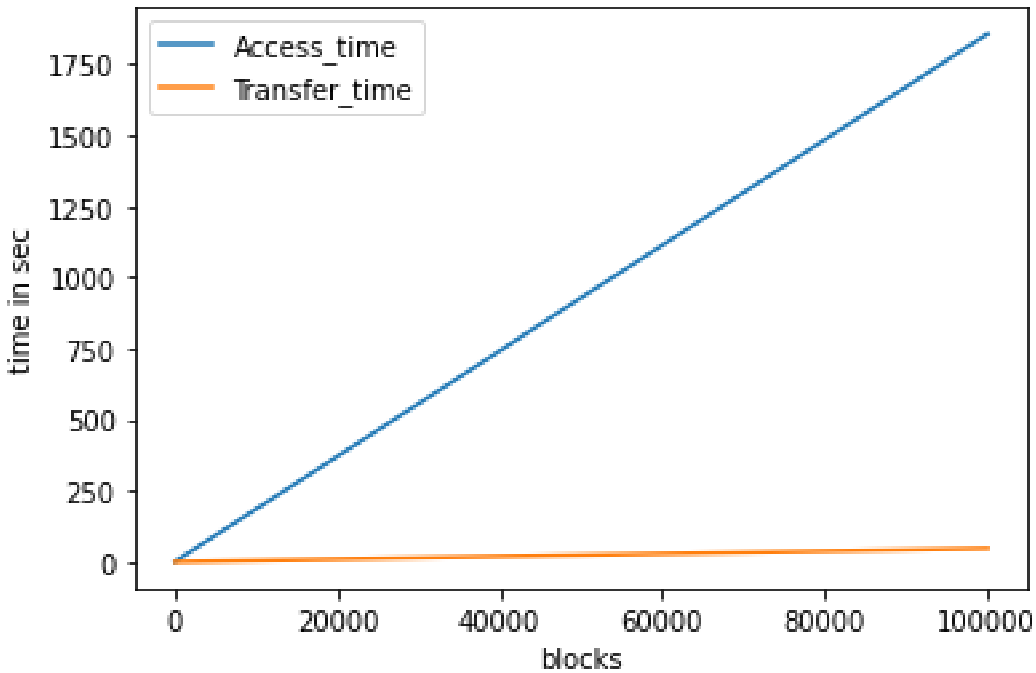
[31]:
```python
def ssd_access_time(blocks):
    return consistent_latency
```

[32]:
```python
def ssd_transfer_time(blocks, block_size):
    return (consistent_latency+ block_size/bandwidth)*blocks
```

[33]:
```python
x=[1,10,100,1000,10000,100000]
y_access=[0]*len(x)
y_transfer=[0]*len(x)
for i in range(len(x)):
    y_access[i]=hdd_access_time(x[i])
    y_transfer[i]=hdd_transfer_time(x[i])
```

[34]:
```
plt.plot(x,y_access)
plt.plot(x,y_transfer)
plt.legend(['Access_time','Transfer_time'])
plt.xlabel('blocks')
plt.ylabel('time in sec')
```

[34]: Text(0, 0.5, 'time in sec')



[ ]:
```
```