

OS LAB: ASSIGNMENT - 6

1 disk.py

Q1.

cmd - python2 disk.py -a 0 -G

Seek time - 0

Rotate time - 165

Transfer time - 30

Total - 195

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 0 -G
OPTIONS seed 0
OPTIONS addr 0
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute False
OPTIONS graphics True
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

WARNING: Setting compute flag to True, as graphics are on

REQUESTS ['0']

Block: 0 Seek: 0 Rotate:165 Transfer: 30 Total: 195
TOTALS Seek: 0 Rotate:165 Transfer: 30 Total: 195
```

```
cmd - python2 disk.py -a 6 -G  
Seek time - 0  
Rotate time - 345  
Transfer Time - 30  
Total Time - 375
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 6 -G  
OPTIONS seed 0  
OPTIONS addr 6  
OPTIONS addrDesc 5,-1,0  
OPTIONS seekSpeed 1  
OPTIONS rotateSpeed 1  
OPTIONS skew 0  
OPTIONS window -1  
OPTIONS policy FIFO  
OPTIONS compute False  
OPTIONS graphics True  
OPTIONS zoning 30,30,30  
OPTIONS lateAddr -1  
OPTIONS lateAddrDesc 0,-1,0  
  
WARNING: Setting compute flag to True, as graphics are on  
REQUESTS ['6']  
  
Block: 6 Seek: 0 Rotate:345 Transfer: 30 Total: 375  
TOTALS Seek: 0 Rotate:345 Transfer: 30 Total: 375
```

```
cmd - python2 disk.py -a 30 -G  
Seek time - 80  
Rotate time - 265  
Transfer Time - 30  
Total Time - 375
```

```

pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 30 -G
OPTIONS seed 0
OPTIONS addr 30
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute False
OPTIONS graphics True
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

WARNING: Setting compute flag to True, as graphics are on

REQUESTS ['30']

Block: 30 Seek: 80 Rotate:265 Transfer: 30 Total: 375
TOTALS Seek: 80 Rotate:265 Transfer: 30 Total: 375

```

cmd - python2 disk.py -a 7,30,8 -G

Seek time - 160

Rotate time - 545

Transfer Time - 90

Total Time - 795

```

pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 7,30,8 -G
OPTIONS seed 0
OPTIONS addr 7,30,8
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute False
OPTIONS graphics True
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

WARNING: Setting compute flag to True, as graphics are on

REQUESTS ['7', '30', '8']

Block: 7 Seek: 0 Rotate: 15 Transfer: 30 Total: 45
Block: 30 Seek: 80 Rotate:220 Transfer: 30 Total: 330
Block: 8 Seek: 80 Rotate:310 Transfer: 30 Total: 420
TOTALS Seek:160 Rotate:545 Transfer: 90 Total: 795

```

```
cmd - python2 disk.py -a 10,11,12,13 -G  
Seek time - 40  
Rotate time - 425  
Transfer Time - 120  
Total Time - 585
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 10,11,12,13 -G  
OPTIONS seed 0  
OPTIONS addr 10,11,12,13  
OPTIONS addrDesc 5,-1,0  
OPTIONS seekSpeed 1  
OPTIONS rotateSpeed 1  
OPTIONS skew 0  
OPTIONS window -1  
OPTIONS policy FIFO  
OPTIONS compute False  
OPTIONS graphics True  
OPTIONS zoning 30,30,30  
OPTIONS lateAddr -1  
OPTIONS lateAddrDesc 0,-1,0  
  
WARNING: Setting compute flag to True, as graphics are on  
  
REQUESTS ['10', '11', '12', '13']  
  
Block: 10 Seek: 0 Rotate:105 Transfer: 30 Total: 135  
Block: 11 Seek: 0 Rotate: 0 Transfer: 30 Total: 30  
Block: 12 Seek: 40 Rotate:320 Transfer: 30 Total: 390  
Block: 13 Seek: 0 Rotate: 0 Transfer: 30 Total: 30  
  
TOTALS Seek: 40 Rotate:425 Transfer:120 Total: 585
```

Q2.

When the search rate rises, the time decreases until it reaches a particular point. When the search rate is raised further, the time reduces to 435 units and remains steady. The time it takes to seek between the tracks is getting shorter. The entire time is limited by the rotation time. The time increases when the seek rate is reduced. We can see that the time has expanded several times over in the case of a 0.1 seek rate. This happens because the time it takes to locate the proper path is longer than it was previously.

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 7,30,8 -G -S 2
OPTIONS seed 0
OPTIONS addr 7,30,8
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 2
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute False
OPTIONS graphics True
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

WARNING: Setting compute flag to True, as graphics are on
REQUESTS ['7', '30', '8']

Block:  7  Seek:  0  Rotate: 15  Transfer: 30  Total:  45
Block: 30  Seek: 40  Rotate:260  Transfer: 30  Total: 330
Block:  8  Seek: 40  Rotate:350  Transfer: 30  Total: 420

TOTALS      Seek: 80  Rotate:625  Transfer: 90  Total: 795
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 7,30,8 -G -S 4
OPTIONS seed 0
OPTIONS addr 7,30,8
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 4
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute False
OPTIONS graphics True
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

WARNING: Setting compute flag to True, as graphics are on
REQUESTS ['7', '30', '8']

Block:  7  Seek:  0  Rotate: 15  Transfer: 30  Total:  45
Block: 30  Seek: 20  Rotate:280  Transfer: 30  Total: 330
Block:  8  Seek: 20  Rotate: 10  Transfer: 30  Total:  60

TOTALS      Seek: 40  Rotate:305  Transfer: 90  Total: 435
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 7,30,8 -G -S 8
OPTIONS seed 0
OPTIONS addr 7,30,8
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 8
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute False
OPTIONS graphics True
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

WARNING: Setting compute flag to True, as graphics are on

REQUESTS ['7', '30', '8']

Block: 7 Seek: 0 Rotate: 15 Transfer: 30 Total: 45
Block: 30 Seek: 10 Rotate:290 Transfer: 30 Total: 330
Block: 8 Seek: 10 Rotate: 20 Transfer: 30 Total: 60

TOTALS Seek: 20 Rotate:325 Transfer: 90 Total: 435
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 7,30,8 -G -S 10
OPTIONS seed 0
OPTIONS addr 7,30,8
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 10
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute False
OPTIONS graphics True
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

WARNING: Setting compute flag to True, as graphics are on

REQUESTS ['7', '30', '8']

Block: 7 Seek: 0 Rotate: 15 Transfer: 30 Total: 45
Block: 30 Seek: 8 Rotate:292 Transfer: 30 Total: 330
Block: 8 Seek: 8 Rotate: 22 Transfer: 30 Total: 60

TOTALS Seek: 16 Rotate:329 Transfer: 90 Total: 435
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 7,30,8 -G -S 40
OPTIONS seed 0
OPTIONS addr 7,30,8
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 40
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute False
OPTIONS graphics True
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

WARNING: Setting compute flag to True, as graphics are on

REQUESTS ['7', '30', '8']

Block: 7 Seek: 0 Rotate: 15 Transfer: 30 Total: 45
Block: 30 Seek: 2 Rotate:298 Transfer: 30 Total: 330
Block: 8 Seek: 2 Rotate: 28 Transfer: 30 Total: 60

TOTALS      Seek: 4 Rotate:341 Transfer: 90 Total: 435
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 7,30,8 -G -S 0.1
OPTIONS seed 0
OPTIONS addr 7,30,8
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 0.1
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute False
OPTIONS graphics True
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

WARNING: Setting compute flag to True, as graphics are on

REQUESTS ['7', '30', '8']

Block: 7 Seek: 0 Rotate: 15 Transfer: 30 Total: 45
Block: 30 Seek:801 Rotate:219 Transfer: 30 Total:1050
Block: 8 Seek:801 Rotate:309 Transfer: 30 Total:1140

TOTALS      Seek:1602 Rotate:543 Transfer: 90 Total:2235
```

Q3.

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 7,30,8 -c -R 0.1
OPTIONS seed 0
OPTIONS addr 7,30,8
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 0.1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS ['7', '30', '8']

Block:  7  Seek:  0  Rotate:150  Transfer:299  Total: 449
Block: 30  Seek: 80  Rotate:2920  Transfer:301  Total:3301
Block:  8  Seek: 80  Rotate:219   Transfer:300  Total: 599

TOTALS      Seek:160  Rotate:3289  Transfer:900  Total:4349

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 7,30,8 -c -R 0.5
OPTIONS seed 0
OPTIONS addr 7,30,8
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 0.5
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS ['7', '30', '8']

Block:  7  Seek:  0  Rotate: 30  Transfer: 60  Total:  90
Block: 30  Seek: 80  Rotate:520  Transfer: 60  Total: 660
Block:  8  Seek: 80  Rotate:700  Transfer: 60  Total: 840

TOTALS      Seek:160  Rotate:1250  Transfer:180  Total:1590

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 7,30,8 -c -R 0.01
OPTIONS seed 0
OPTIONS addr 7,30,8
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 0.01
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS ['7', '30', '8']

Block: 7 Seek: 0 Rotate:1500 Transfer:3000 Total:4500
Block: 30 Seek: 80 Rotate:29920 Transfer:3001 Total:33001
Block: 8 Seek: 80 Rotate:2920 Transfer:2999 Total:5999

TOTALS Seek:160 Rotate:34340 Transfer:9000 Total:43500
pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

Rotation times and transfer times rise when rotation rates are lowered.

As a consequence, we can see that the time required is significantly longer than in earlier instances.

Q4.

```
pranav@pranav: ~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 7,30,8 -c -p FIFO
OPTIONS seed 0
OPTIONS addr 7,30,8
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS ['7', '30', '8']

Block:  7  Seek:  0  Rotate: 15  Transfer: 30  Total:  45
Block: 30  Seek: 80  Rotate:220  Transfer: 30  Total: 330
Block:  8  Seek: 80  Rotate:310  Transfer: 30  Total: 420

TOTALS      Seek:160  Rotate:545  Transfer: 90  Total: 795

pranav@pranav: ~/Desktop/OS_lab/os/assignment2$
```

```
pranav@pranav: ~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 7,30,8 -c -p SSTF
OPTIONS seed 0
OPTIONS addr 7,30,8
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy SSTF
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS ['7', '30', '8']

Block:  7  Seek:  0  Rotate: 15  Transfer: 30  Total:  45
Block:  8  Seek:  0  Rotate:  0  Transfer: 30  Total:  30
Block: 30  Seek: 80  Rotate:190  Transfer: 30  Total: 300

TOTALS      Seek: 80  Rotate:205  Transfer: 90  Total: 375

pranav@pranav: ~/Desktop/OS_lab/os/assignment2$
```

The request that comes first is handled first in the FIFO system. As a result, the processing order is 7,30,8.

For the SSTF, the request with the least seek time is handled first. As a result, the processing order is 7,8,30.

Q5.

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 7,30,8 -c -p SATF
OPTIONS seed 0
OPTIONS addr 7,30,8
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy SATF
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS ['7', '30', '8']

Block:  7  Seek:  0  Rotate: 15  Transfer: 30  Total:  45
Block:  8  Seek:  0  Rotate:  0  Transfer: 30  Total:  30
Block: 30  Seek: 80  Rotate:190  Transfer: 30  Total: 300

TOTALS      Seek: 80  Rotate:205  Transfer: 90  Total: 375

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 7,30,8 -c -p SSTF
OPTIONS seed 0
OPTIONS addr 7,30,8
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy SSTF
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS ['7', '30', '8']

Block: 7 Seek: 0 Rotate: 15 Transfer: 30 Total: 45
Block: 8 Seek: 0 Rotate: 0 Transfer: 30 Total: 30
Block: 30 Seek: 80 Rotate:190 Transfer: 30 Total: 300

TOTALS Seek: 80 Rotate:205 Transfer: 90 Total: 375

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

No, the SATF makes no effect for workloads 7,30,8.

Let us now attempt to create a set of requests in which the SATF outperforms the SSTF.

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 32,4 -c -p SATF -S 40 -R 2
OPTIONS seed 0
OPTIONS addr 32,4
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 40
OPTIONS rotateSpeed 2
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy SATF
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS ['32', '4']

Block: 32 Seek: 2 Rotate: 20 Transfer: 15 Total: 37
Block: 4 Seek: 2 Rotate:103 Transfer: 15 Total: 120

TOTALS Seek: 4 Rotate:123 Transfer: 30 Total: 157

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 32,4 -c -p SSTF -S 40 -R 2
OPTIONS seed 0
OPTIONS addr 32,4
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 40
OPTIONS rotateSpeed 2
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy SSTF
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS ['32', '4']

Block: 4 Seek: 0 Rotate:142 Transfer: 15 Total: 157
Block: 32 Seek: 2 Rotate: 43 Transfer: 15 Total: 60
TOTALS Seek: 2 Rotate:185 Transfer: 30 Total: 217

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

We can see that SATF outperformed SSTF in this case.

Q6.

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 10,11,12,13 -c
OPTIONS seed 0
OPTIONS addr 10,11,12,13
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS ['10', '11', '12', '13']

Block: 10 Seek: 0 Rotate:105 Transfer: 30 Total: 135
Block: 11 Seek: 0 Rotate: 0 Transfer: 30 Total: 30
Block: 12 Seek: 40 Rotate:320 Transfer: 30 Total: 390
Block: 13 Seek: 0 Rotate: 0 Transfer: 30 Total: 30
TOTALS Seek: 40 Rotate:425 Transfer:120 Total: 585

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

The problem is that the seek must be moved to the inner track when the number 12 must be read. The 12 goes ahead of the arm as it advances to the inner track, thus we'll have to wait for a full circuit to get at 12.

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 10,11,12,13 -o 2 -c
OPTIONS seed 0
OPTIONS addr 10,11,12,13
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 2
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS ['10', '11', '12', '13']

Block: 10 Seek: 0 Rotate:105 Transfer: 30 Total: 135
Block: 11 Seek: 0 Rotate: 0 Transfer: 30 Total: 30
Block: 12 Seek: 40 Rotate: 20 Transfer: 30 Total: 90
Block: 13 Seek: 0 Rotate: 0 Transfer: 30 Total: 30

TOTALS Seek: 40 Rotate:125 Transfer:120 Total: 285
pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 10,11,12,13 -o 3 -c
OPTIONS seed 0
OPTIONS addr 10,11,12,13
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 3
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS ['10', '11', '12', '13']

Block: 10 Seek: 0 Rotate:105 Transfer: 30 Total: 135
Block: 11 Seek: 0 Rotate: 0 Transfer: 30 Total: 30
Block: 12 Seek: 40 Rotate: 50 Transfer: 30 Total: 120
Block: 13 Seek: 0 Rotate: 0 Transfer: 30 Total: 30

TOTALS Seek: 40 Rotate:155 Transfer:120 Total: 315
pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

We enhanced the workload performance by including the skew. The best skew that has been discovered is 2 (-o 2). The overall duration grows as the number of skew values increases, as does the number of skew values smaller than two.

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 10,11,12,13 -c -S 2
OPTIONS seed 0
OPTIONS addr 10,11,12,13
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 2
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS ['10', '11', '12', '13']

Block: 10 Seek: 0 Rotate:105 Transfer: 30 Total: 135
Block: 11 Seek: 0 Rotate: 0 Transfer: 30 Total: 30
Block: 12 Seek: 20 Rotate:340 Transfer: 30 Total: 390
Block: 13 Seek: 0 Rotate: 0 Transfer: 30 Total: 30

TOTALS Seek: 20 Rotate:445 Transfer:120 Total: 585
pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 10,11,12,13 -c -S 4
OPTIONS seed 0
OPTIONS addr 10,11,12,13
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 4
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS ['10', '11', '12', '13']

Block: 10 Seek: 0 Rotate:105 Transfer: 30 Total: 135
Block: 11 Seek: 0 Rotate: 0 Transfer: 30 Total: 30
Block: 12 Seek: 10 Rotate:350 Transfer: 30 Total: 390
Block: 13 Seek: 0 Rotate: 0 Transfer: 30 Total: 30

TOTALS Seek: 10 Rotate:455 Transfer:120 Total: 585
pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

Because the time used is restricted by the time it takes for the disc to revolve, the time taken remains constant even if the seek rate is raised.

The Formula for the skew is the mentioned below:

$$\text{skew} = \text{ceil}((40/2)*1/30) = 1$$

$$\text{skew} = \text{ceil}((40/4)*1/30) = 1$$

Q7.

Different seeds used are : 3, 2, 1

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -z 10,20,30 -a -1 -A 5,-1,0 -c
OPTIONS seed 0
OPTIONS addr -1
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 10,20,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS [45, 40, 22, 13, 27]

Block: 45  Seek: 40  Rotate:310  Transfer: 20  Total: 370
Block: 40  Seek: 0   Rotate:240  Transfer: 20  Total: 260
Block: 22  Seek: 40  Rotate: 85  Transfer: 10  Total: 135
Block: 13  Seek: 0   Rotate:260  Transfer: 10  Total: 270
Block: 27  Seek: 0   Rotate:130  Transfer: 10  Total: 140

TOTALS      Seek: 80  Rotate:1025  Transfer: 70  Total:1175

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

$$\text{Outer Bandwidth} = 3/(135+270+140)=0.0055$$

$$\text{Middle Bandwidth}=2/(370+260)=0.0032$$

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -z 10,20,30 -s 3 -a -1 -A 5,-1,0 -c
OPTIONS seed 3
OPTIONS addr -1
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 10,20,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS [12, 29, 19, 32, 33]

Block: 12 Seek: 0 Rotate:295 Transfer: 10 Total: 305
Block: 29 Seek: 0 Rotate:160 Transfer: 10 Total: 170
Block: 19 Seek: 0 Rotate:250 Transfer: 10 Total: 260
Block: 32 Seek: 0 Rotate:120 Transfer: 10 Total: 130
Block: 33 Seek: 0 Rotate: 0 Transfer: 10 Total: 10

TOTALS      Seek: 0 Rotate:825 Transfer: 50 Total: 875

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

Outer Bandwidth=5/875=0.0057

Middle Bandwidth=0

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -z 10,20,30 -s 2 -a -1 -A 5,-1,0 -c
OPTIONS seed 2
OPTIONS addr -1
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 10,20,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS [51, 51, 3, 4, 45]

Block: 51 Seek: 40 Rotate: 70 Transfer: 20 Total: 130
Block: 51 Seek: 0 Rotate:340 Transfer: 20 Total: 360
Block: 3 Seek: 40 Rotate: 35 Transfer: 10 Total: 85
Block: 4 Seek: 0 Rotate: 0 Transfer: 10 Total: 10
Block: 45 Seek: 40 Rotate: 85 Transfer: 20 Total: 145

TOTALS      Seek:120 Rotate:530 Transfer: 80 Total: 730

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

Outer Bandwidth = 2/(85+10) = 0.0211

Middle Bandwidth = 3/(130+360+145) = 0.0047

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -z 10,20,30 -s 1 -a -1 -A 5,-1,0 -c
OPTIONS seed 1
OPTIONS addr -1
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 10,20,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS [7, 45, 41, 13, 26]

Block: 7 Seek: 0 Rotate:245 Transfer: 10 Total: 255
Block: 45 Seek: 40 Rotate: 55 Transfer: 20 Total: 115
Block: 41 Seek: 0 Rotate:260 Transfer: 20 Total: 280
Block: 13 Seek: 40 Rotate:335 Transfer: 10 Total: 385
Block: 26 Seek: 0 Rotate:120 Transfer: 10 Total: 130

TOTALS      Seek: 80 Rotate:1015 Transfer: 70 Total:1165

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

Outer Bandwidth = $3/(255+385+130)=0.0039$

Middle Bandwidth = $2/(115+280)=0.0051$

Q8.

I had taken the values as 1, 100, 500 and 1000 for the scheduling windows in the commands.

Cmd: `python2 disk.py -A 1000,-1,0 -p SATF -w 1 -c`

```
TOTALS      Seek:20960 Rotate:169165 Transfer:30000 Total:220125
```

Cmd: `python2 disk.py -A 1000,-1,0 -p SATF -w 100 -c`

```
TOTALS      Seek:1440 Rotate:5835 Transfer:30000 Total:37275
```

Cmd: `python2 disk.py -A 1000,-1,0 -p SATF -w 500 -c`

```
TOTALS      Seek:1280 Rotate:4195 Transfer:30000 Total:35475
```

Cmd: `python2 disk.py -A 1000,-1,0 -p SATF -w 1000 -c`

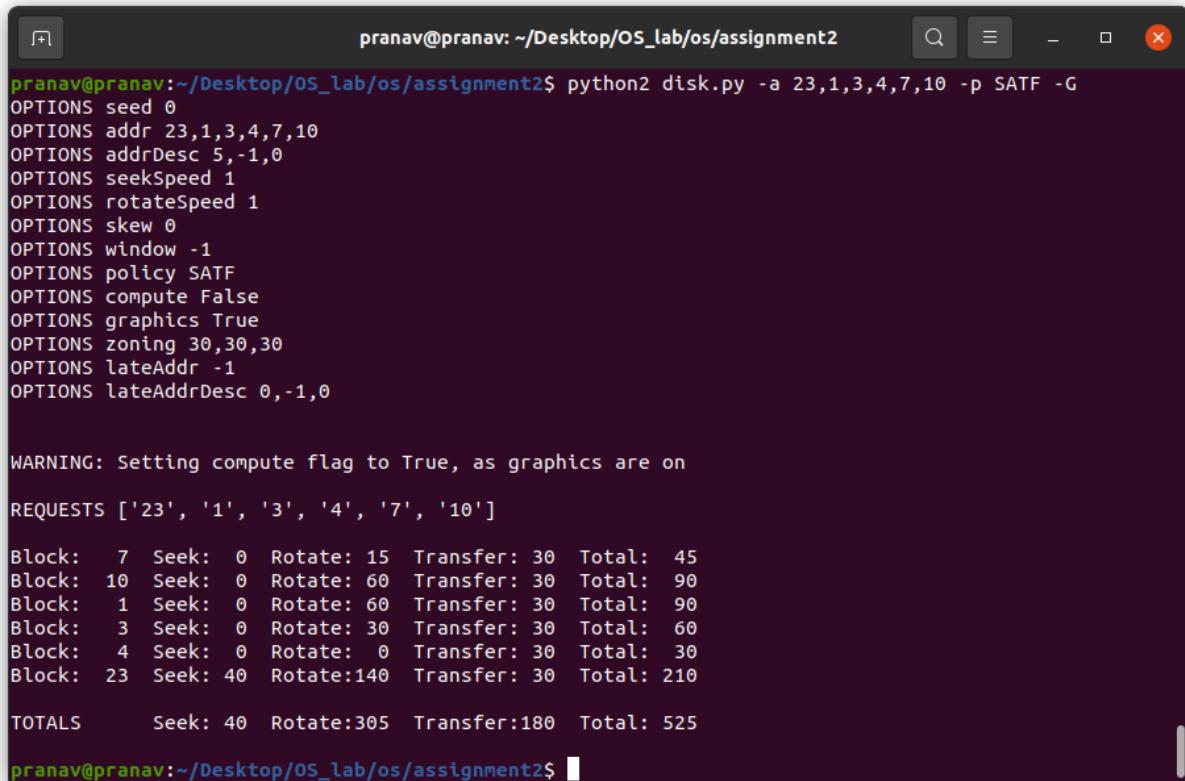
```
TOTALS      Seek:1520 Rotate:3955 Transfer:30000 Total:35475
```

To maximize performance, the size of the scheduling window is adjusted to match the size of the disc.

The policy becomes First In First Out (FIFO) when the scheduling window size is set to one.

Q9.

I have chosen 23 from the second outer track and 1,3,4,7,10 from the outermost track.



A screenshot of a terminal window titled "pranav@pranav: ~/Desktop/OS_lab/os/assignment2". The terminal displays the output of a Python script named "disk.py" with the command-line arguments "-a 23,1,3,4,7,10 -p SATF -G". The output shows various configuration options like seed, address, seek speed, and rotation speed, followed by a warning about compute being True when graphics are on. It then lists the requests: [23, 1, 3, 4, 7, 10]. The script then processes these requests using the SATF policy, showing detailed metrics for each block (Seek, Rotate, Transfer, Total time) and finally summing up the totals.

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 23,1,3,4,7,10 -p SATF -G
OPTIONS seed 0
OPTIONS addr 23,1,3,4,7,10
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy SATF
OPTIONS compute False
OPTIONS graphics True
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

WARNING: Setting compute flag to True, as graphics are on

REQUESTS ['23', '1', '3', '4', '7', '10']

Block:  7  Seek:  0  Rotate: 15  Transfer: 30  Total:  45
Block: 10  Seek:  0  Rotate: 60  Transfer: 30  Total:  90
Block:  1  Seek:  0  Rotate: 60  Transfer: 30  Total:  90
Block:  3  Seek:  0  Rotate: 30  Transfer: 30  Total:  60
Block:  4  Seek:  0  Rotate:  0  Transfer: 30  Total:  30
Block: 23  Seek: 40  Rotate:140  Transfer: 30  Total: 210

TOTALS      Seek: 40  Rotate:305  Transfer:180  Total: 525
pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

The SATF, as seen in the screenshot above, denies the request for 23, which was the original request. This is due to the fact that other queries have the fastest access time.

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 23,1,3,4,7,10 -p BSATF -w 4 -G
OPTIONS seed 0
OPTIONS addr 23,1,3,4,7,10
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window 4
OPTIONS policy BSATF
OPTIONS compute False
OPTIONS graphics True
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

WARNING: Setting compute flag to True, as graphics are on

REQUESTS ['23', '1', '3', '4', '7', '10']

Block: 23 Seek: 40 Rotate: 95 Transfer: 30 Total: 165
Block: 3 Seek: 40 Rotate: 50 Transfer: 30 Total: 120
Block: 4 Seek: 0 Rotate: 0 Transfer: 30 Total: 30
Block: 1 Seek: 0 Rotate:240 Transfer: 30 Total: 270
Block: 7 Seek: 0 Rotate:150 Transfer: 30 Total: 180
Block: 10 Seek: 0 Rotate: 60 Transfer: 30 Total: 90

TOTALS      Seek: 80 Rotate:595 Transfer:180 Total: 855

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

Request 23 was served, even though it was at the end of the timeframe, according to the BSATF. The BSATF, as we can see, overcomes the problem of famine. There is no starvation since all requests in the window size are fulfilled first, followed by requests from the next windows.

We can see that the SATF performed marginally better than the BSATF, but it did not alleviate the starving problem. We may modify the size of the scheduling window and pick the window size that delivers the best performance to make the tradeoff between performance and hunger avoidance.

Q10.

Set of requests in which greedy is not optimal:

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 9,20 -c
OPTIONS seed 0
OPTIONS addr 9,20
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy FIFO
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

REQUESTS ['9', '20']

Block:  9  Seek:  0  Rotate: 75  Transfer: 30  Total: 105
Block: 20  Seek: 40  Rotate:260  Transfer: 30  Total: 330

TOTALS      Seek: 40  Rotate:335  Transfer: 60  Total: 435
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 disk.py -a 9,20 -c -p SATF
OPTIONS seed 0
OPTIONS addr 9,20
OPTIONS addrDesc 5,-1,0
OPTIONS seekSpeed 1
OPTIONS rotateSpeed 1
OPTIONS skew 0
OPTIONS window -1
OPTIONS policy SATF
OPTIONS compute True
OPTIONS graphics False
OPTIONS zoning 30,30,30
OPTIONS lateAddr -1
OPTIONS lateAddrDesc 0,-1,0

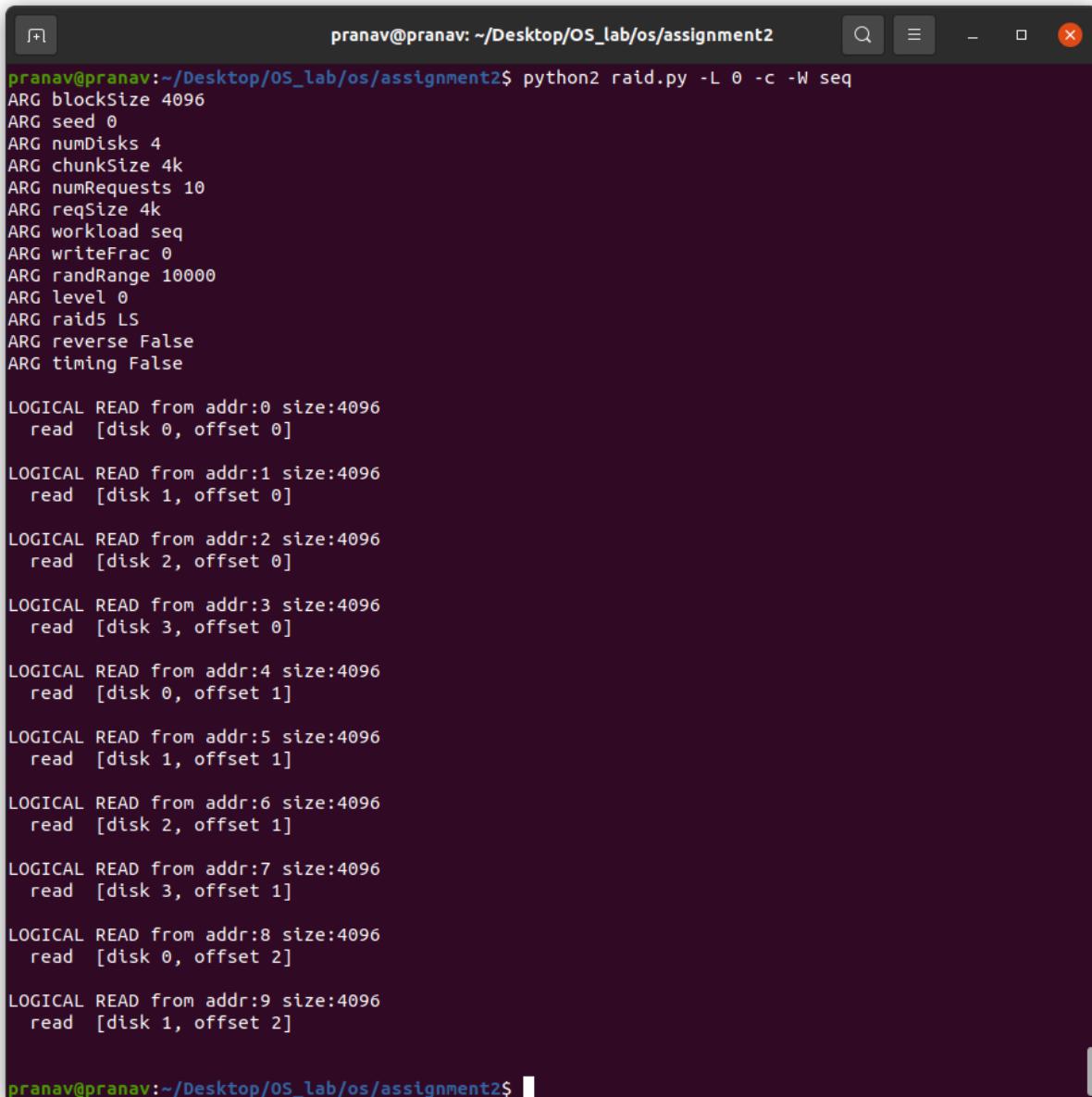
REQUESTS ['9', '20']

Block: 20  Seek: 40  Rotate: 5  Transfer: 30  Total: 75
Block:  9  Seek: 40  Rotate:320  Transfer: 30  Total: 390

TOTALS      Seek: 80  Rotate:325  Transfer: 60  Total: 465
```

2 raid.py

Q1.



The screenshot shows a terminal window with the following content:

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 0 -c -W seq
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 10
ARG reqSize 4k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 0
ARG raid5 LS
ARG reverse False
ARG timing False

LOGICAL READ from addr:0 size:4096
  read [disk 0, offset 0]

LOGICAL READ from addr:1 size:4096
  read [disk 1, offset 0]

LOGICAL READ from addr:2 size:4096
  read [disk 2, offset 0]

LOGICAL READ from addr:3 size:4096
  read [disk 3, offset 0]

LOGICAL READ from addr:4 size:4096
  read [disk 0, offset 1]

LOGICAL READ from addr:5 size:4096
  read [disk 1, offset 1]

LOGICAL READ from addr:6 size:4096
  read [disk 2, offset 1]

LOGICAL READ from addr:7 size:4096
  read [disk 3, offset 1]

LOGICAL READ from addr:8 size:4096
  read [disk 0, offset 2]

LOGICAL READ from addr:9 size:4096
  read [disk 1, offset 2]

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

Disk 0 : {0, 4, 8}, Disk 1 : {1, 5, 9}, Disk 2 : {2, 6}, Disk 3 : {3, 7}

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 1 -c -W seq
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 10
ARG reqSize 4k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 1
ARG raid5 LS
ARG reverse False
ARG timing False

LOGICAL READ from addr:0 size:4096
  read [disk 0, offset 0]

LOGICAL READ from addr:1 size:4096
  read [disk 2, offset 0]

LOGICAL READ from addr:2 size:4096
  read [disk 1, offset 1]

LOGICAL READ from addr:3 size:4096
  read [disk 3, offset 1]

LOGICAL READ from addr:4 size:4096
  read [disk 0, offset 2]

LOGICAL READ from addr:5 size:4096
  read [disk 2, offset 2]

LOGICAL READ from addr:6 size:4096
  read [disk 1, offset 3]

LOGICAL READ from addr:7 size:4096
  read [disk 3, offset 3]

LOGICAL READ from addr:8 size:4096
  read [disk 0, offset 4]

LOGICAL READ from addr:9 size:4096
  read [disk 2, offset 4]

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

Disk 0 : {0, 4, 8}, Disk 1 : {2, 6}, Disk 2 : {1, 5, 9}, Disk 3 : {3, 7}

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 4 -c -W seq
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 10
ARG reqSize 4k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 4
ARG raid5 LS
ARG reverse False
ARG timing False

LOGICAL READ from addr:0 size:4096
  read [disk 0, offset 0]
LOGICAL READ from addr:1 size:4096
  read [disk 1, offset 0]
LOGICAL READ from addr:2 size:4096
  read [disk 2, offset 0]
LOGICAL READ from addr:3 size:4096
  read [disk 0, offset 1]
LOGICAL READ from addr:4 size:4096
  read [disk 1, offset 1]
LOGICAL READ from addr:5 size:4096
  read [disk 2, offset 1]
LOGICAL READ from addr:6 size:4096
  read [disk 0, offset 2]
LOGICAL READ from addr:7 size:4096
  read [disk 1, offset 2]
LOGICAL READ from addr:8 size:4096
  read [disk 2, offset 2]
LOGICAL READ from addr:9 size:4096
  read [disk 0, offset 3]
```

Disk 0 : {0, 3, 6, 9}, Disk 1 : {1, 4, 7}, Disk 2 : {2, 5, 8}

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 5 -S LS -c -W seq
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 10
ARG reqSize 4k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 5
ARG raid5 LS
ARG reverse False
ARG timing False

LOGICAL READ from addr:0 size:4096
  read [disk 0, offset 0]
LOGICAL READ from addr:1 size:4096
  read [disk 1, offset 0]
LOGICAL READ from addr:2 size:4096
  read [disk 2, offset 0]
LOGICAL READ from addr:3 size:4096
  read [disk 3, offset 1]
LOGICAL READ from addr:4 size:4096
  read [disk 0, offset 1]
LOGICAL READ from addr:5 size:4096
  read [disk 1, offset 1]
LOGICAL READ from addr:6 size:4096
  read [disk 2, offset 2]
LOGICAL READ from addr:7 size:4096
  read [disk 3, offset 2]
LOGICAL READ from addr:8 size:4096
  read [disk 0, offset 2]
LOGICAL READ from addr:9 size:4096
  read [disk 1, offset 3]
```

Left Symmetric Disk 0 : {0, 4, 8, P4}, Disk 1 : {1, 5, P3, 9},

Disk 2 : {2, P2, 6}, Disk 3 : {P1, 3, 7}

```

pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 5 -S 5 LA -c -W seq
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 10
ARG reqSize 4k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 5
ARG raid5 LA
ARG reverse False
ARG timing False

LOGICAL READ from addr:0 size:4096
  read [disk 0, offset 0]
LOGICAL READ from addr:1 size:4096
  read [disk 1, offset 0]
LOGICAL READ from addr:2 size:4096
  read [disk 2, offset 0]
LOGICAL READ from addr:3 size:4096
  read [disk 0, offset 1]
LOGICAL READ from addr:4 size:4096
  read [disk 1, offset 1]
LOGICAL READ from addr:5 size:4096
  read [disk 3, offset 1]
LOGICAL READ from addr:6 size:4096
  read [disk 0, offset 2]
LOGICAL READ from addr:7 size:4096
  read [disk 2, offset 2]
LOGICAL READ from addr:8 size:4096
  read [disk 3, offset 2]
LOGICAL READ from addr:9 size:4096
  read [disk 1, offset 3]

```

Left Asymmetric Disk 0 : {0, 3, 6, P4}, Disk 1 : {1, 4, P3, 9},

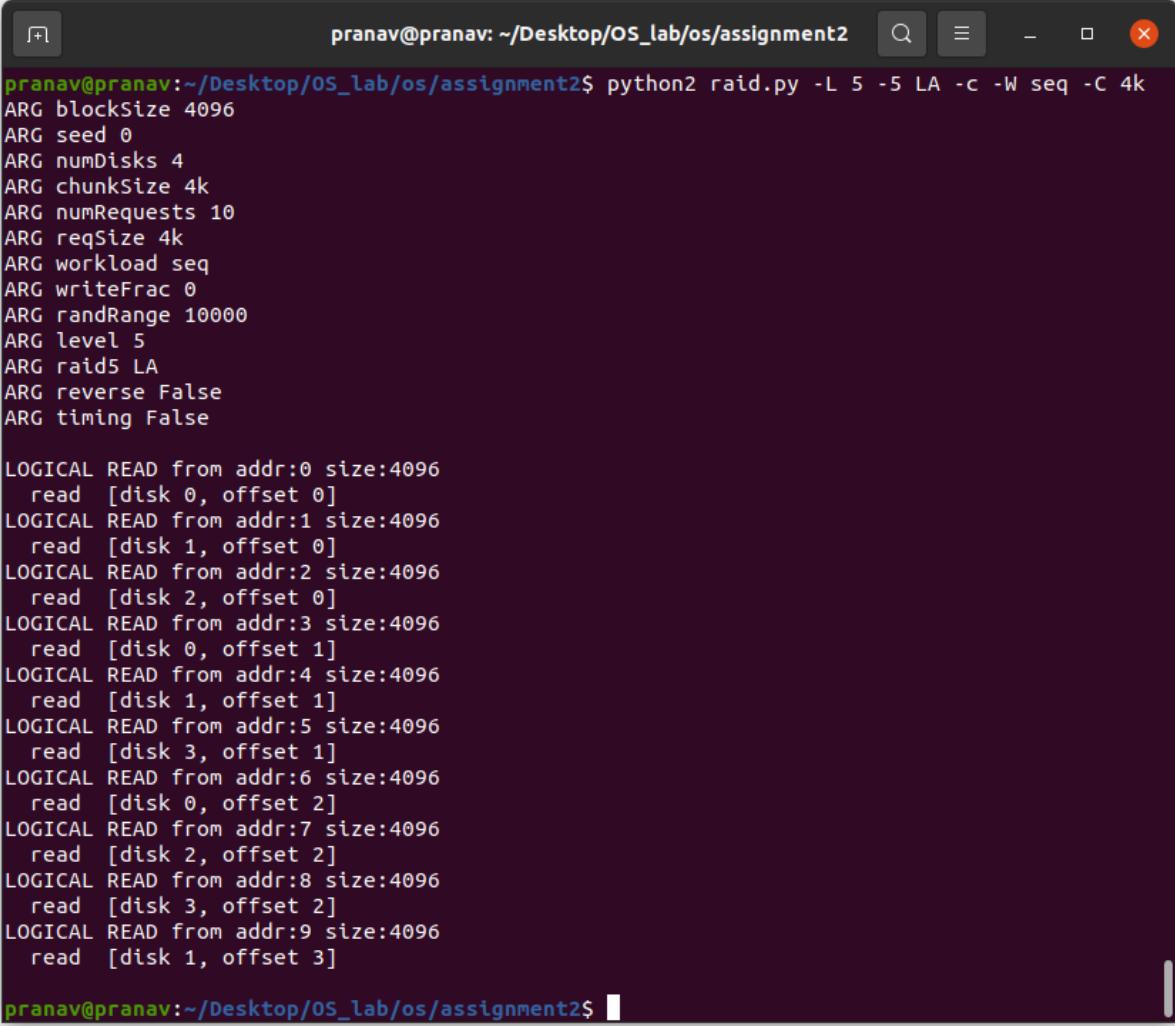
Disk 2 : {2, P2, 7}, Disk 3 : {P1, 5, 8}

The mapping does not change on changing the seeds.

The difference between symmetric and asymmetric may be summarized as follows: The data strips in asymmetric RAIDs ignore parity and skip it until they reach the next available space. Data strips in symmetric RAIDs are handled differently; whenever they meet a parity block, they shift sideways and down to the next stripe set.

Q2.

I changed the chuck size as: -C 4k, -C 8k, -C 16k



pranav@pranav:~/Desktop/OS_lab/os/assignment2\$ python2 raid.py -L 5 -5 LA -c -W seq -C 4k
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 10
ARG reqSize 4k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 5
ARG raid5 LA
ARG reverse False
ARG timing False

LOGICAL READ from addr:0 size:4096
read [disk 0, offset 0]
LOGICAL READ from addr:1 size:4096
read [disk 1, offset 0]
LOGICAL READ from addr:2 size:4096
read [disk 2, offset 0]
LOGICAL READ from addr:3 size:4096
read [disk 0, offset 1]
LOGICAL READ from addr:4 size:4096
read [disk 1, offset 1]
LOGICAL READ from addr:5 size:4096
read [disk 3, offset 1]
LOGICAL READ from addr:6 size:4096
read [disk 0, offset 2]
LOGICAL READ from addr:7 size:4096
read [disk 2, offset 2]
LOGICAL READ from addr:8 size:4096
read [disk 3, offset 2]
LOGICAL READ from addr:9 size:4096
read [disk 1, offset 3]

pranav@pranav:~/Desktop/OS_lab/os/assignment2\$

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 5 -5 LA -c -W seq -C 8k
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 8k
ARG numRequests 10
ARG reqSize 4k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 5
ARG raid5 LA
ARG reverse False
ARG timing False

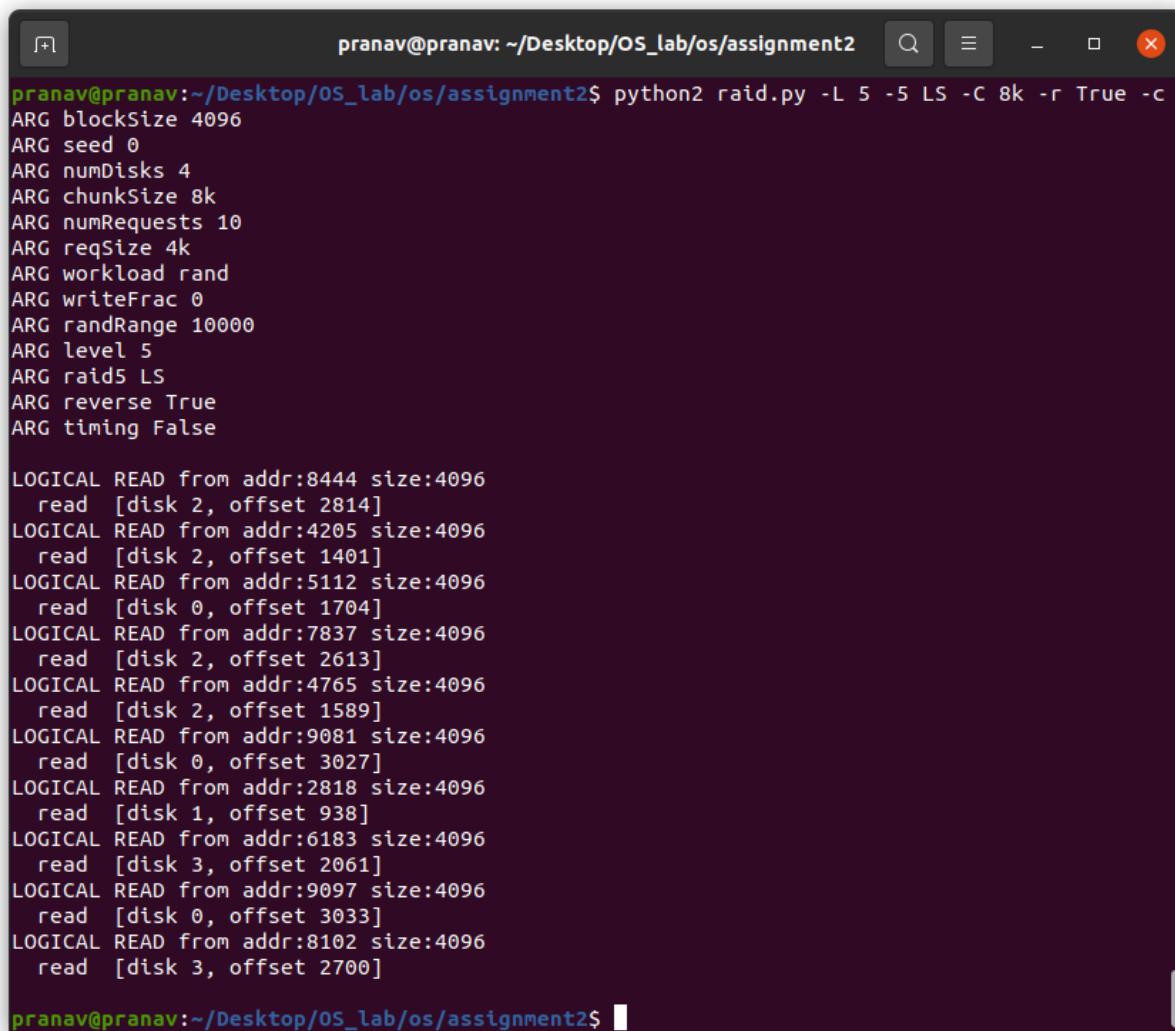
LOGICAL READ from addr:0 size:4096
read [disk 0, offset 0]
LOGICAL READ from addr:1 size:4096
read [disk 0, offset 1]
LOGICAL READ from addr:2 size:4096
read [disk 1, offset 0]
LOGICAL READ from addr:3 size:4096
read [disk 1, offset 1]
LOGICAL READ from addr:4 size:4096
read [disk 2, offset 0]
LOGICAL READ from addr:5 size:4096
read [disk 2, offset 1]
LOGICAL READ from addr:6 size:4096
read [disk 0, offset 2]
LOGICAL READ from addr:7 size:4096
read [disk 0, offset 3]
LOGICAL READ from addr:8 size:4096
read [disk 1, offset 2]
LOGICAL READ from addr:9 size:4096
read [disk 1, offset 3]
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 5 -5 LA -c -W seq -C 16k
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 16k
ARG numRequests 10
ARG reqSize 4k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 5
ARG raid5 LA
ARG reverse False
ARG timing False

LOGICAL READ from addr:0 size:4096
read [disk 0, offset 0]
LOGICAL READ from addr:1 size:4096
read [disk 0, offset 1]
LOGICAL READ from addr:2 size:4096
read [disk 0, offset 2]
LOGICAL READ from addr:3 size:4096
read [disk 0, offset 3]
LOGICAL READ from addr:4 size:4096
read [disk 1, offset 0]
LOGICAL READ from addr:5 size:4096
read [disk 1, offset 1]
LOGICAL READ from addr:6 size:4096
read [disk 1, offset 2]
LOGICAL READ from addr:7 size:4096
read [disk 1, offset 3]
LOGICAL READ from addr:8 size:4096
read [disk 2, offset 0]
LOGICAL READ from addr:9 size:4096
read [disk 2, offset 1]
```

From the above outputs we can say that, using a bigger chunk puts multiple parity blocks in the same disks for level 5.

Q3.



A terminal window titled "pranav@pranav: ~/Desktop/OS_lab/os/assignment2\$". The window contains the following text:

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 5 -5 LS -C 8k -r True -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 8k
ARG numRequests 10
ARG reqSize 4k
ARG workload rand
ARG writeFrac 0
ARG randRange 10000
ARG level 5
ARG raid5 LS
ARG reverse True
ARG timing False

LOGICAL READ from addr:8444 size:4096
    read [disk 2, offset 2814]
LOGICAL READ from addr:4205 size:4096
    read [disk 2, offset 1401]
LOGICAL READ from addr:5112 size:4096
    read [disk 0, offset 1704]
LOGICAL READ from addr:7837 size:4096
    read [disk 2, offset 2613]
LOGICAL READ from addr:4765 size:4096
    read [disk 2, offset 1589]
LOGICAL READ from addr:9081 size:4096
    read [disk 0, offset 3027]
LOGICAL READ from addr:2818 size:4096
    read [disk 1, offset 938]
LOGICAL READ from addr:6183 size:4096
    read [disk 3, offset 2061]
LOGICAL READ from addr:9097 size:4096
    read [disk 0, offset 3033]
LOGICAL READ from addr:8102 size:4096
    read [disk 3, offset 2700]

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 5 -S LA -C 8k -r True -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 8k
ARG numRequests 10
ARG reqSize 4k
ARG workload rand
ARG writeFrac 0
ARG randRange 10000
ARG level 5
ARG raid5 LA
ARG reverse True
ARG timing False

LOGICAL READ from addr:8444 size:4096
  read [disk 2, offset 2814]
LOGICAL READ from addr:4205 size:4096
  read [disk 2, offset 1401]
LOGICAL READ from addr:5112 size:4096
  read [disk 0, offset 1704]
LOGICAL READ from addr:7837 size:4096
  read [disk 0, offset 2613]
LOGICAL READ from addr:4765 size:4096
  read [disk 0, offset 1589]
LOGICAL READ from addr:9081 size:4096
  read [disk 1, offset 3027]
LOGICAL READ from addr:2818 size:4096
  read [disk 3, offset 938]
LOGICAL READ from addr:6183 size:4096
  read [disk 2, offset 2061]
LOGICAL READ from addr:9097 size:4096
  read [disk 0, offset 3033]
LOGICAL READ from addr:8102 size:4096
  read [disk 2, offset 2700]

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

Q4.

Raid 4:

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 4 -S 4k -c -W seq -r
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 10
ARG reqSize 4k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 4
ARG raid5 LS
ARG reverse True
ARG timing False

LOGICAL READ from addr:0 size:4096
  read [disk 0, offset 0]
LOGICAL READ from addr:1 size:4096
  read [disk 1, offset 0]
LOGICAL READ from addr:2 size:4096
  read [disk 2, offset 0]
LOGICAL READ from addr:3 size:4096
  read [disk 0, offset 1]
LOGICAL READ from addr:4 size:4096
  read [disk 1, offset 1]
LOGICAL READ from addr:5 size:4096
  read [disk 2, offset 1]
LOGICAL READ from addr:6 size:4096
  read [disk 0, offset 2]
LOGICAL READ from addr:7 size:4096
  read [disk 1, offset 2]
LOGICAL READ from addr:8 size:4096
  read [disk 2, offset 2]
LOGICAL READ from addr:9 size:4096
  read [disk 0, offset 3]

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 4 -S 8k -c -W seq -r
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 10
ARG reqSize 8k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 4
ARG raid5 LS
ARG reverse True
ARG timing False

LOGICAL READ from addr:0 size:8192
  read [disk 0, offset 0]      read [disk 1, offset 0]
LOGICAL READ from addr:2 size:8192
  read [disk 2, offset 0]      read [disk 0, offset 1]
LOGICAL READ from addr:4 size:8192
  read [disk 1, offset 1]      read [disk 2, offset 1]
LOGICAL READ from addr:6 size:8192
  read [disk 0, offset 2]      read [disk 1, offset 2]
LOGICAL READ from addr:8 size:8192
  read [disk 2, offset 2]      read [disk 0, offset 3]
LOGICAL READ from addr:10 size:8192
  read [disk 1, offset 3]      read [disk 2, offset 3]
LOGICAL READ from addr:12 size:8192
  read [disk 0, offset 4]      read [disk 1, offset 4]
LOGICAL READ from addr:14 size:8192
  read [disk 2, offset 4]      read [disk 0, offset 5]
LOGICAL READ from addr:16 size:8192
  read [disk 1, offset 5]      read [disk 2, offset 5]
LOGICAL READ from addr:18 size:8192
  read [disk 0, offset 6]      read [disk 1, offset 6]

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 4 -S 12k -c -W seq -r
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 10
ARG reqSize 12k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 4
ARG raid5 LS
ARG reverse True
ARG timing False

LOGICAL READ from addr:0 size:12288
    read [disk 0, offset 0]    read [disk 1, offset 0]    read [disk 2, offset 0]
LOGICAL READ from addr:3 size:12288
    read [disk 0, offset 1]    read [disk 1, offset 1]    read [disk 2, offset 1]
LOGICAL READ from addr:6 size:12288
    read [disk 0, offset 2]    read [disk 1, offset 2]    read [disk 2, offset 2]
LOGICAL READ from addr:9 size:12288
    read [disk 0, offset 3]    read [disk 1, offset 3]    read [disk 2, offset 3]
LOGICAL READ from addr:12 size:12288
    read [disk 0, offset 4]    read [disk 1, offset 4]    read [disk 2, offset 4]
LOGICAL READ from addr:15 size:12288
    read [disk 0, offset 5]    read [disk 1, offset 5]    read [disk 2, offset 5]
LOGICAL READ from addr:18 size:12288
    read [disk 0, offset 6]    read [disk 1, offset 6]    read [disk 2, offset 6]
LOGICAL READ from addr:21 size:12288
    read [disk 0, offset 7]    read [disk 1, offset 7]    read [disk 2, offset 7]
LOGICAL READ from addr:24 size:12288
    read [disk 0, offset 8]    read [disk 1, offset 8]    read [disk 2, offset 8]
LOGICAL READ from addr:27 size:12288
    read [disk 0, offset 9]    read [disk 1, offset 9]    read [disk 2, offset 9]

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 4 -S 16k -c -W seq -r
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 10
ARG reqSize 16k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 4
ARG raid5 LS
ARG reverse True
ARG timing False

LOGICAL READ from addr:0 size:16384
    read [disk 0, offset 0]    read [disk 1, offset 0]    read [disk 2, offset 0]    read [disk 0, offset 1]
LOGICAL READ from addr:4 size:16384
    read [disk 1, offset 1]    read [disk 2, offset 1]    read [disk 0, offset 2]    read [disk 1, offset 2]
LOGICAL READ from addr:8 size:16384
    read [disk 2, offset 2]    read [disk 0, offset 3]    read [disk 1, offset 3]    read [disk 2, offset 3]
LOGICAL READ from addr:12 size:16384
    read [disk 0, offset 4]    read [disk 1, offset 4]    read [disk 2, offset 4]    read [disk 0, offset 5]
LOGICAL READ from addr:16 size:16384
    read [disk 1, offset 5]    read [disk 2, offset 5]    read [disk 0, offset 6]    read [disk 1, offset 6]
LOGICAL READ from addr:20 size:16384
    read [disk 2, offset 6]    read [disk 0, offset 7]    read [disk 1, offset 7]    read [disk 2, offset 7]
LOGICAL READ from addr:24 size:16384
    read [disk 0, offset 8]    read [disk 1, offset 8]    read [disk 2, offset 8]    read [disk 0, offset 9]
LOGICAL READ from addr:28 size:16384
    read [disk 1, offset 9]    read [disk 2, offset 9]    read [disk 0, offset 10]   read [disk 1, offset 10]
LOGICAL READ from addr:32 size:16384
    read [disk 2, offset 10]   read [disk 0, offset 11]   read [disk 1, offset 11]   read [disk 2, offset 11]
LOGICAL READ from addr:36 size:16384
    read [disk 0, offset 12]   read [disk 1, offset 12]   read [disk 2, offset 12]   read [disk 0, offset 13]

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

Raid 5:

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 5 -S 4k -c -W seq -r
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 10
ARG reqSize 4k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 5
ARG raids5 LS
ARG reverse True
ARG timing False

LOGICAL READ from addr:0 size:4096
  read [disk 0, offset 0]
LOGICAL READ from addr:1 size:4096
  read [disk 1, offset 0]
LOGICAL READ from addr:2 size:4096
  read [disk 2, offset 0]
LOGICAL READ from addr:3 size:4096
  read [disk 3, offset 1]
LOGICAL READ from addr:4 size:4096
  read [disk 0, offset 1]
LOGICAL READ from addr:5 size:4096
  read [disk 1, offset 1]
LOGICAL READ from addr:6 size:4096
  read [disk 2, offset 2]
LOGICAL READ from addr:7 size:4096
  read [disk 3, offset 2]
LOGICAL READ from addr:8 size:4096
  read [disk 0, offset 2]
LOGICAL READ from addr:9 size:4096
  read [disk 1, offset 3]

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 5 -S 8k -c -W seq -r
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 10
ARG reqSize 8k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 5
ARG raids5 LS
ARG reverse True
ARG timing False

LOGICAL READ from addr:0 size:8192
  read [disk 0, offset 0]    read [disk 1, offset 0]
LOGICAL READ from addr:2 size:8192
  read [disk 2, offset 0]    read [disk 3, offset 1]
LOGICAL READ from addr:4 size:8192
  read [disk 0, offset 1]    read [disk 1, offset 1]
LOGICAL READ from addr:6 size:8192
  read [disk 2, offset 2]    read [disk 3, offset 2]
LOGICAL READ from addr:8 size:8192
  read [disk 0, offset 2]    read [disk 1, offset 3]
LOGICAL READ from addr:10 size:8192
  read [disk 2, offset 3]    read [disk 3, offset 3]
LOGICAL READ from addr:12 size:8192
  read [disk 0, offset 4]    read [disk 1, offset 4]
LOGICAL READ from addr:14 size:8192
  read [disk 2, offset 4]    read [disk 3, offset 5]
LOGICAL READ from addr:16 size:8192
  read [disk 0, offset 5]    read [disk 1, offset 5]
LOGICAL READ from addr:18 size:8192
  read [disk 2, offset 6]    read [disk 3, offset 6]

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 5 -S 12k -c -W seq -r
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 10
ARG reqSize 12k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 5
ARG raids5 LS
ARG reverse True
ARG timing False

LOGICAL READ from addr:0 size:12288
    read [disk 0, offset 0]    read [disk 1, offset 0]    read [disk 2, offset 0]
LOGICAL READ from addr:3 size:12288
    read [disk 3, offset 1]    read [disk 0, offset 1]    read [disk 1, offset 1]
LOGICAL READ from addr:6 size:12288
    read [disk 2, offset 2]    read [disk 3, offset 2]    read [disk 0, offset 2]
LOGICAL READ from addr:9 size:12288
    read [disk 1, offset 3]    read [disk 2, offset 3]    read [disk 3, offset 3]
LOGICAL READ from addr:12 size:12288
    read [disk 0, offset 4]    read [disk 1, offset 4]    read [disk 2, offset 4]
LOGICAL READ from addr:15 size:12288
    read [disk 3, offset 5]    read [disk 0, offset 5]    read [disk 1, offset 5]
LOGICAL READ from addr:18 size:12288
    read [disk 2, offset 6]    read [disk 3, offset 6]    read [disk 0, offset 6]
LOGICAL READ from addr:21 size:12288
    read [disk 1, offset 7]    read [disk 2, offset 7]    read [disk 3, offset 7]
LOGICAL READ from addr:24 size:12288
    read [disk 0, offset 8]    read [disk 1, offset 8]    read [disk 2, offset 8]
LOGICAL READ from addr:27 size:12288
    read [disk 3, offset 9]    read [disk 0, offset 9]    read [disk 1, offset 9]

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 5 -S 16k -c -W seq -r
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 10
ARG reqSize 16k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 5
ARG raids5 LS
ARG reverse True
ARG timing False

LOGICAL READ from addr:0 size:16384
    read [disk 0, offset 0]    read [disk 1, offset 0]    read [disk 2, offset 0]    read [disk 3, offset 1]
LOGICAL READ from addr:4 size:16384
    read [disk 0, offset 1]    read [disk 1, offset 1]    read [disk 2, offset 2]    read [disk 3, offset 2]
LOGICAL READ from addr:8 size:16384
    read [disk 0, offset 2]    read [disk 1, offset 3]    read [disk 2, offset 3]    read [disk 3, offset 3]
LOGICAL READ from addr:12 size:16384
    read [disk 0, offset 4]    read [disk 1, offset 4]    read [disk 2, offset 4]    read [disk 3, offset 5]
LOGICAL READ from addr:16 size:16384
    read [disk 0, offset 5]    read [disk 1, offset 5]    read [disk 2, offset 6]    read [disk 3, offset 6]
LOGICAL READ from addr:20 size:16384
    read [disk 0, offset 6]    read [disk 1, offset 7]    read [disk 2, offset 7]    read [disk 3, offset 7]
LOGICAL READ from addr:24 size:16384
    read [disk 0, offset 8]    read [disk 1, offset 8]    read [disk 2, offset 8]    read [disk 3, offset 9]
LOGICAL READ from addr:28 size:16384
    read [disk 0, offset 9]    read [disk 1, offset 9]    read [disk 2, offset 10]   read [disk 3, offset 10]
LOGICAL READ from addr:32 size:16384
    read [disk 0, offset 10]   read [disk 1, offset 11]   read [disk 2, offset 11]   read [disk 3, offset 11]
LOGICAL READ from addr:36 size:16384
    read [disk 0, offset 12]   read [disk 1, offset 12]   read [disk 2, offset 12]   read [disk 3, offset 13]

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

As the request size increases in size, the read occurs from adjacent discs as well. As a result, we'll have to do more disc accesses, which will increase I/O time. As data is stored on the next disc, the number of disc accesses for each request grows. For big request sizes, RAID 4 and

RAID 5 are significantly more efficient since data reads are faster due to parity and striping, and data is typically distributed among all discs, making it more lucrative in those circumstances.

Thus, for 16k requests, RAID 4 and RAID 5 are much more I/O efficient.

Q5.

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 0 -n 100 -t -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload rand
ARG writeFrac 0
ARG randRange 10000
ARG level 0
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 100.00 I/Os:    28 (sequential:0 nearly:1 random:27)
disk:1 busy:  93.91 I/Os:    29 (sequential:0 nearly:6 random:23)
disk:2 busy:  87.92 I/Os:    24 (sequential:0 nearly:0 random:24)
disk:3 busy:  65.94 I/Os:    19 (sequential:0 nearly:1 random:18)

STAT totalTime 275.7
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 1 -n 100 -t -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload rand
ARG writeFrac 0
ARG randRange 10000
ARG level 1
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 100.00 I/Os:    28 (sequential:0 nearly:1 random:27)
disk:1 busy:  86.98 I/Os:    24 (sequential:0 nearly:0 random:24)
disk:2 busy:  97.52 I/Os:    29 (sequential:0 nearly:3 random:26)
disk:3 busy:  65.23 I/Os:    19 (sequential:0 nearly:1 random:18)

STAT totalTime 278.7
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 4 -n 100 -t -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload rand
ARG writeFrac 0
ARG randRange 10000
ARG level 4
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 78.48 I/Os: 30 (sequential:0 nearly:0 random:30)
disk:1 busy: 100.00 I/Os: 40 (sequential:0 nearly:3 random:37)
disk:2 busy: 76.46 I/Os: 30 (sequential:0 nearly:2 random:28)
disk:3 busy: 0.00 I/Os: 0 (sequential:0 nearly:0 random:0)

STAT totalTime 386.1
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 5 -n 100 -t -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload rand
ARG writeFrac 0
ARG randRange 10000
ARG level 5
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 100.00 I/Os: 28 (sequential:0 nearly:1 random:27)
disk:1 busy: 95.84 I/Os: 29 (sequential:0 nearly:5 random:24)
disk:2 busy: 87.60 I/Os: 24 (sequential:0 nearly:0 random:24)
disk:3 busy: 65.70 I/Os: 19 (sequential:0 nearly:1 random:18)

STAT totalTime 276.7
```

The overall timings are as follows: 275.7 for RAID 0, 278.7 for RAID 1, 386.1 for RAID 4, and 276.7 for RAID 5.

RAID 0 is the quickest of the options, while RAID 5 is quicker than RAID 4.

RAID 5 also has quicker writes than RAID 1, therefore it's comparable to RAID 1.

Q6.

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 0 -t -n 100 -c -D 8
ARG blockSize 4096
ARG seed 0
ARG numDisks 8
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload rand
ARG writeFrac 0
ARG randRange 10000
ARG level 0
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 67.86 I/Os: 12 (sequential:0 nearly:3 random:9)
disk:1 busy: 63.58 I/Os: 12 (sequential:0 nearly:3 random:9)
disk:2 busy: 75.46 I/Os: 13 (sequential:0 nearly:3 random:10)
disk:3 busy: 33.35 I/Os: 6 (sequential:0 nearly:1 random:5)
disk:4 busy: 95.65 I/Os: 16 (sequential:0 nearly:2 random:14)
disk:5 busy: 100.00 I/Os: 17 (sequential:0 nearly:3 random:14)
disk:6 busy: 70.03 I/Os: 11 (sequential:0 nearly:1 random:10)
disk:7 busy: 77.44 I/Os: 13 (sequential:0 nearly:1 random:12)

STAT totalTime 156.5
```

Performance ratio = $275.7 / 156.5 = 1.76$

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 1 -t -n 100 -c -D 8
ARG blockSize 4096
ARG seed 0
ARG numDisks 8
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload rand
ARG writeFrac 0
ARG randRange 10000
ARG level 1
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 67.76 I/Os: 12 (sequential:0 nearly:1 random:11)
disk:1 busy: 92.07 I/Os: 16 (sequential:0 nearly:1 random:15)
disk:2 busy: 64.36 I/Os: 12 (sequential:0 nearly:2 random:10)
disk:3 busy: 100.00 I/Os: 17 (sequential:0 nearly:1 random:16)
disk:4 busy: 77.47 I/Os: 13 (sequential:0 nearly:1 random:12)
disk:5 busy: 66.21 I/Os: 11 (sequential:0 nearly:0 random:11)
disk:6 busy: 32.12 I/Os: 6 (sequential:0 nearly:1 random:5)
disk:7 busy: 72.23 I/Os: 13 (sequential:0 nearly:1 random:12)

STAT totalTime 167.8
```

Performance ratio = $278.7 / 167.8 = 1.66$

```

pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 4 -t -n 100 -c -D 8
ARG blockSize 4096
ARG seed 0
ARG numDisks 8
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload rand
ARG writeFrac 0
ARG randRange 10000
ARG level 4
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 94.00 I/Os: 17 (sequential:0 nearly:2 random:15)
disk:1 busy: 66.61 I/Os: 12 (sequential:0 nearly:2 random:10)
disk:2 busy: 100.00 I/Os: 18 (sequential:0 nearly:3 random:15)
disk:3 busy: 72.36 I/Os: 13 (sequential:0 nearly:2 random:11)
disk:4 busy: 76.73 I/Os: 13 (sequential:0 nearly:1 random:12)
disk:5 busy: 78.30 I/Os: 13 (sequential:0 nearly:1 random:12)
disk:6 busy: 83.70 I/Os: 14 (sequential:0 nearly:1 random:13)
disk:7 busy: 0.00 I/Os: 0 (sequential:0 nearly:0 random:0)

STAT totalTime 165.0

```

Performance ratio = 386.1 / 165.0 = 2.34

```

pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 5 -t -n 100 -c -D 8
ARG blockSize 4096
ARG seed 0
ARG numDisks 8
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload rand
ARG writeFrac 0
ARG randRange 10000
ARG level 5
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 68.35 I/Os: 12 (sequential:0 nearly:3 random:9)
disk:1 busy: 63.49 I/Os: 12 (sequential:0 nearly:3 random:9)
disk:2 busy: 76.04 I/Os: 13 (sequential:0 nearly:3 random:10)
disk:3 busy: 33.04 I/Os: 6 (sequential:0 nearly:1 random:5)
disk:4 busy: 95.15 I/Os: 16 (sequential:0 nearly:2 random:14)
disk:5 busy: 100.00 I/Os: 17 (sequential:0 nearly:3 random:14)
disk:6 busy: 69.86 I/Os: 11 (sequential:0 nearly:1 random:10)
disk:7 busy: 76.42 I/Os: 13 (sequential:0 nearly:1 random:12)

STAT totalTime 158.6

```

Performance ratio = 276.5 / 158.6 = 1.74

Q7.

For RAID 0 :

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 0 -n 100 -t -w 100 -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload rand
ARG writeFrac 100
ARG randRange 10000
ARG level 0
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 100.00 I/Os: 28 (sequential:0 nearly:1 random:27)
disk:1 busy: 93.91 I/Os: 29 (sequential:0 nearly:6 random:23)
disk:2 busy: 87.92 I/Os: 24 (sequential:0 nearly:0 random:24)
disk:3 busy: 65.94 I/Os: 19 (sequential:0 nearly:1 random:18)

STAT totalTime 275.7
```

Total time taken is 275.7

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 0 -n 100 -t -w 100 -D 8 -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 8
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload rand
ARG writeFrac 100
ARG randRange 10000
ARG level 0
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 67.86 I/Os: 12 (sequential:0 nearly:3 random:9)
disk:1 busy: 63.58 I/Os: 12 (sequential:0 nearly:3 random:9)
disk:2 busy: 75.46 I/Os: 13 (sequential:0 nearly:3 random:10)
disk:3 busy: 33.35 I/Os: 6 (sequential:0 nearly:1 random:5)
disk:4 busy: 95.65 I/Os: 16 (sequential:0 nearly:2 random:14)
disk:5 busy: 100.00 I/Os: 17 (sequential:0 nearly:3 random:14)
disk:6 busy: 70.03 I/Os: 11 (sequential:0 nearly:1 random:10)
disk:7 busy: 77.44 I/Os: 13 (sequential:0 nearly:1 random:12)

STAT totalTime 156.5
```

Total time taken is 156.5

Performance scale up = $275.7/156.5 = 1.761$.

To do a rough estimate of the time taken we have 100 accesses and 4disks. So, let us take an average of 10 units of time for each write then the estimate becomes $100*10/4$ equals 250 that is a nearest value to 275.7.

For RAID 4:

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 4 -n 100 -t -w 100 -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload rand
ARG writeFrac 100
ARG randRange 10000
ARG level 4
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 30.84 I/Os: 60 (sequential:0 nearly:30 random:30)
disk:1 busy: 39.30 I/Os: 80 (sequential:0 nearly:43 random:37)
disk:2 busy: 30.05 I/Os: 60 (sequential:0 nearly:32 random:28)
disk:3 busy: 100.00 I/Os: 200 (sequential:0 nearly:107 random:93)

STAT totalTime 982.5
```

Total time taken is 982.5

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 4 -n 100 -t -w 100 -D 8 -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 8
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload rand
ARG writeFrac 100
ARG randRange 10000
ARG level 4
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 16.54 I/Os: 34 (sequential:0 nearly:19 random:15)
disk:1 busy: 11.72 I/Os: 24 (sequential:0 nearly:14 random:10)
disk:2 busy: 17.59 I/Os: 36 (sequential:0 nearly:21 random:15)
disk:3 busy: 12.73 I/Os: 26 (sequential:0 nearly:15 random:11)
disk:4 busy: 13.50 I/Os: 26 (sequential:0 nearly:14 random:12)
disk:5 busy: 13.78 I/Os: 26 (sequential:0 nearly:14 random:12)
disk:6 busy: 14.73 I/Os: 28 (sequential:0 nearly:15 random:13)
disk:7 busy: 100.00 I/Os: 200 (sequential:0 nearly:113 random:87)

STAT totalTime 937.8
```

Total time taken is 937.8

Performance scale up = $982.5/937.8 = 1.047$.

Here we have parity involved and the writes are expensive so the estimation cannot be done linearly as above.

For RAID 5:

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 5 -n 100 -t -w 100 -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload rand
ARG writeFrac 100
ARG randRange 10000
ARG level 5
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 99.32 I/Os: 100 (sequential:0 nearly:53 random:47)
disk:1 busy: 96.02 I/Os: 100 (sequential:0 nearly:55 random:45)
disk:2 busy: 99.62 I/Os: 100 (sequential:0 nearly:52 random:48)
disk:3 busy: 100.00 I/Os: 100 (sequential:0 nearly:53 random:47)

STAT totalTime 497.4
```

Total time taken is 497.4

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 5 -n 100 -t -w 100 -D 8 -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 8
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload rand
ARG writeFrac 100
ARG randRange 10000
ARG level 5
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 87.90 I/Os: 56 (sequential:0 nearly:33 random:23)
disk:1 busy: 58.95 I/Os: 40 (sequential:0 nearly:26 random:14)
disk:2 busy: 63.05 I/Os: 40 (sequential:0 nearly:23 random:17)
disk:3 busy: 72.91 I/Os: 42 (sequential:0 nearly:21 random:21)
disk:4 busy: 99.66 I/Os: 64 (sequential:0 nearly:37 random:27)
disk:5 busy: 85.60 I/Os: 54 (sequential:0 nearly:33 random:21)
disk:6 busy: 69.44 I/Os: 44 (sequential:0 nearly:26 random:18)
disk:7 busy: 100.00 I/Os: 60 (sequential:1 nearly:31 random:28)

STAT totalTime 290.9
```

Total time taken is 290.9

Performance scale up = $497.4/290.9 = 1.709$.

Here we have parity involved and the writes are expensive so the estimation cannot be done linearly as above.

Q8.

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 0 -n 100 -t -W seq -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 0
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 100.00 I/Os: 25 (sequential:24 nearly:0 random:1)
disk:1 busy: 100.00 I/Os: 25 (sequential:24 nearly:0 random:1)
disk:2 busy: 100.00 I/Os: 25 (sequential:24 nearly:0 random:1)
disk:3 busy: 100.00 I/Os: 25 (sequential:24 nearly:0 random:1)

STAT totalTime 12.5
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 0 -n 100 -t -w 100 -W seq -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload seq
ARG writeFrac 100
ARG randRange 10000
ARG level 0
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 100.00 I/Os: 25 (sequential:24 nearly:0 random:1)
disk:1 busy: 100.00 I/Os: 25 (sequential:24 nearly:0 random:1)
disk:2 busy: 100.00 I/Os: 25 (sequential:24 nearly:0 random:1)
disk:3 busy: 100.00 I/Os: 25 (sequential:24 nearly:0 random:1)

STAT totalTime 12.5
```

Both take equal amounts of time. Thus we can see that there is no performance variance with sequential workload for read and writes.

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 1 -n 100 -t -W seq -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 1
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 100.00 I/Os: 25 (sequential:0 nearly:24 random:1)
disk:1 busy: 100.00 I/Os: 25 (sequential:0 nearly:24 random:1)
disk:2 busy: 100.00 I/Os: 25 (sequential:0 nearly:24 random:1)
disk:3 busy: 100.00 I/Os: 25 (sequential:0 nearly:24 random:1)

STAT totalTime 14.9

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 1 -n 100 -t -w 100 -W seq -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload seq
ARG writeFrac 100
ARG randRange 10000
ARG level 1
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 100.00 I/Os: 50 (sequential:49 nearly:0 random:1)
disk:1 busy: 100.00 I/Os: 50 (sequential:49 nearly:0 random:1)
disk:2 busy: 100.00 I/Os: 50 (sequential:49 nearly:0 random:1)
disk:3 busy: 100.00 I/Os: 50 (sequential:49 nearly:0 random:1)

STAT totalTime 15.0

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

Again both take around the same time. Here we can see that there is no performance variance with sequential workload for read and writes.

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 4 -n 100 -t -W seq -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 4
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 100.00 I/Os: 34 (sequential:33 nearly:0 random:1)
disk:1 busy: 99.25 I/Os: 33 (sequential:32 nearly:0 random:1)
disk:2 busy: 99.25 I/Os: 33 (sequential:32 nearly:0 random:1)
disk:3 busy: 0.00 I/Os: 0 (sequential:0 nearly:0 random:0)

STAT totalTime 13.4
pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 4 -n 100 -t -w 100 -W seq -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload seq
ARG writeFrac 100
ARG randRange 10000
ARG level 4
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 100.00 I/Os: 68 (sequential:33 nearly:34 random:1)
disk:1 busy: 99.25 I/Os: 66 (sequential:32 nearly:33 random:1)
disk:2 busy: 99.25 I/Os: 66 (sequential:32 nearly:33 random:1)
disk:3 busy: 100.00 I/Os: 200 (sequential:33 nearly:166 random:1)

STAT totalTime 13.4
pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

No difference in time. Thus we can see that there is no performance variance with sequential workload for read and writes.

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 5 -n 100 -t -W seq -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload seq
ARG writeFrac 0
ARG randRange 10000
ARG level 5
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 100.00 I/Os: 25 (sequential:16 nearly:8 random:1)
disk:1 busy: 100.00 I/Os: 25 (sequential:16 nearly:8 random:1)
disk:2 busy: 100.00 I/Os: 25 (sequential:16 nearly:8 random:1)
disk:3 busy: 100.00 I/Os: 25 (sequential:16 nearly:8 random:1)

STAT totalTime 13.3

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

```
pranav@pranav:~/Desktop/OS_lab/os/assignment2$ python2 raid.py -L 5 -n 100 -t -w 100 -W seq -c
ARG blockSize 4096
ARG seed 0
ARG numDisks 4
ARG chunkSize 4k
ARG numRequests 100
ARG reqSize 4k
ARG workload seq
ARG writeFrac 100
ARG randRange 10000
ARG level 5
ARG raid5 LS
ARG reverse False
ARG timing True

disk:0 busy: 99.25 I/Os: 98 (sequential:32 nearly:65 random:1)
disk:1 busy: 99.25 I/Os: 98 (sequential:32 nearly:65 random:1)
disk:2 busy: 100.00 I/Os: 100 (sequential:33 nearly:66 random:1)
disk:3 busy: 100.00 I/Os: 104 (sequential:33 nearly:70 random:1)

STAT totalTime 13.4

pranav@pranav:~/Desktop/OS_lab/os/assignment2$
```

Again similar amounts of time. Thus we can see that there is no performance variance with sequential workload for read and writes for a given request size.

Conclusion:

The total time for different request sizes, such as 8k, 12k, and 16k, also increases as the workload request size increases since we may have to access more discs for processing a request. However, with workloads of 12k and 16k, RAID4 and RAID 5 are often preferable. Because they scale well in certain situations. This is also true for sequential workloads, therefore RAID 4,5 is preferable for larger workloads with request sizes such as 12k and 16k.