

CS 342 - Operating Systems Project 2

Question 3)

I created a FileCreator program as explained in the README.txt file. After checking the validity of the programs with various files that I created, by using FileCreator, I created a randomly constructed files. I tested my syn_thistogram program by using these files and my findings in terms of run-time are below :

Some conventions:

Real means the real time elapsed in terms of universal time like clock time.

User means the time elapsed as CPU seconds in the user mode.

Sys means the time elapsed as CPU seconds in the kernel mode.

The timing format is *m*.***s, the asterisk number before m represents minutes and the number before s represents the seconds.

1st Experiment: File sizes are 50.000, minvalue: 1, maxvalue: 100.000, number of files: 4, bincount: 50

1st Iteration: Batch size(B) : 5

```
real  0m0.060s
user  0m0.055s
sys   0m0.020s
```

2nd Iteration: Batch size(B) : 10

```
real  0m0.063s
user  0m0.064s
sys   0m0.012s
```

3rd Iteration: Batch size(B) : 20

```
real  0m0.061s
user  0m0.060s
sys   0m0.015s
```

4th Iteration: Batch size(B) : 50

```
real  0m0.059s
user  0m0.049s
sys   0m0.017s
```

5th Iteration: Batch size(B) : 100

```
real    0m0.057s
user    0m0.064s
sys     0m0.000s
```

2nd Experiment: File sizes are 100.000, minvalue: 1, maxvalue: 100.000, number of files: 4, bincount: 50

1st Iteration: Batch size(B) : 5

```
real    0m0.113s
user    0m0.083s
sys     0m0.053s
```

2nd Iteration: Batch size(B) : 10

```
real    0m0.111s
user    0m0.103s
sys     0m0.019s
```

3rd Iteration: Batch size(B) : 20

```
real    0m0.095s
user    0m0.108s
sys     0m0.017s
```

4th Iteration: Batch size(B) : 50

```
real    0m0.097s
user    0m0.105s
sys     0m0.004s
```

5th Iteration: Batch size(B) : 100

```
real    0m0.096s
user    0m0.079s
sys     0m0.034s
```

3rd Experiment: File sizes are 1.000.000, minvalue: 1, maxvalue: 100.000, number of files: 4, bincount: 50

1st Iteration: Batch size(B) : 5

```
real    0m0.942s
user    0m1.210s
sys     0m0.135s
```

2nd Iteration: Batch size(B) : 10

```
real    0m1.025s
user    0m1.178s
sys     0m0.145s
```

3rd Iteration: Batch size(B) : 20

```
real    0m0.977s
user    0m1.095s
sys     0m0.271s
```

4th Iteration: Batch size(B) : 50

```
real    0m0.902s
user    0m0.990s
sys     0m0.094s
```

5th Iteration: Batch size(B) : 100

```
real    0m0.884s
user    0m0.953s
sys     0m0.111s
```

4th Experiment: File sizes are 100.000, minvalue: 1, maxvalue: 100.000, number of files: 6, bincount: 50

1st Iteration: Batch size(B) : 5

```
real    0m0.139s
user    0m0.154s
sys     0m0.013s
```

2nd Iteration: Batch size(B) : 10

```
real    0m0.135s
user    0m0.122s
sys     0m0.047s
```

3rd Iteration: Batch size(B) : 20

```
real    0m0.132s
user    0m0.128s
sys     0m0.032s
```

4th Iteration: Batch size(B) : 50

```
real    0m0.132s
user    0m0.128s
sys     0m0.032s
```

5th Iteration: Batch size(B) : 100

```
real    0m0.137s
user    0m0.123s
sys     0m0.020s
```

5th Experiment: File sizes are 1.000.000, minvalue: 1, maxvalue: 100.000, number of files: 6, bincount: 50

1st Iteration: Batch size(B) : 5

```
real    0m1.145s
user    0m1.368s
sys     0m0.176s
```

2nd Iteration: Batch size(B) : 10

```
real    0m1.080s
user    0m1.343s
sys     0m0.087s
```

3rd Iteration: Batch size(B) : 20

```
real    0m1.077s
user    0m1.228s
sys     0m0.119s
```

4th Iteration: Batch size(B) : 50

```
real    0m1.209s
user    0m1.202s
sys     0m0.090s
```

5th Iteration: Batch size(B) : 100

```
real    0m1.141s
user    0m1.202s
sys     0m0.102s
```

6th Experiment: File sizes are 100.000, minvalue: 1, maxvalue: 100.000, number of files: 8, bincount: 50

1st Iteration: Batch size(B) : 5

```
real    0m0.172s
user    0m0.196s
sys     0m0.018s
```

2nd Iteration: Batch size(B) : 10

```
real    0m0.180s
user    0m0.174s
sys     0m0.043s
```

3rd Iteration: Batch size(B) : 20

```
real    0m0.178s
user    0m0.164s
sys     0m0.033s
```

4th Iteration: Batch size(B) : 50

```
real    0m0.174s
user    0m0.174s
sys     0m0.013s
```

5th Iteration: Batch size(B) : 100

```
real    0m0.180s
user    0m0.153s
sys     0m0.036s
```

7th Experiment: File sizes are 1.000.000, minvalue: 1, maxvalue: 100.000, number of files: 8, bincount: 50

1st Iteration: Batch size(B) : 5

```
real    0m1.964s
user    0m2.657s
sys     0m0.324s
```

2nd Iteration: Batch size(B) : 10

```
real    0m1.489s
user    0m1.821s
sys     0m0.188s
```

3rd Iteration: Batch size(B) : 20

```
real    0m1.477s
user    0m1.647s
sys     0m0.105s
```

4th Iteration: Batch size(B) : 50

```
real    0m1.642s
user    0m1.579s
sys     0m0.154s
```

5th Iteration: Batch size(B) : 100

real	0m1.527s
user	0m1.584s
sys	0m0.119s

Conclusion :

According to the datas that are gathered from the above experiments, for smaller size of inputs like the 1st experiment(50.000) and 2nd experiment(100.000), bigger batch size seems to make program slightly faster. However, the difference between the batch size 5 to 100 for those 2 experiments are very small. While the input size gets bigger through the 1st experiment to 3rd experiment that are with the same number of files, batch size increment is observed to have more effect on runtime.

A second sequence of experiments which are 4-5th experiments are conducted to analyze the impact of file numbers on the program runtime. This time the same experiment sequence are conducted with 6 files. The result shows that batch size incrimination does not affect the runtimes that much. Both of the experiments are resulted into negligible runtime impact like 0.002 seconds. After this finding, another set of experiments(6-7) are made with 8 files another different input sizes in order to observe the relationship between file number and batch size.

In conclusion, above findings suggest that while file number increases, if those files have smaller sizes, batch size increase affects runtime in a negligible way. However, when file sizes gets bigger, batch size's effect on runtime becomes more visible. So, while the number of files that are being processed and their relative sizes get bigger, bigger batch sizes result into better runtime performance in terms of turnaround time.