

A1 Part I: Disk access characteristics

Liliya Aliyeva

Sequential write to file

1. Plot of write data rate versus block size.

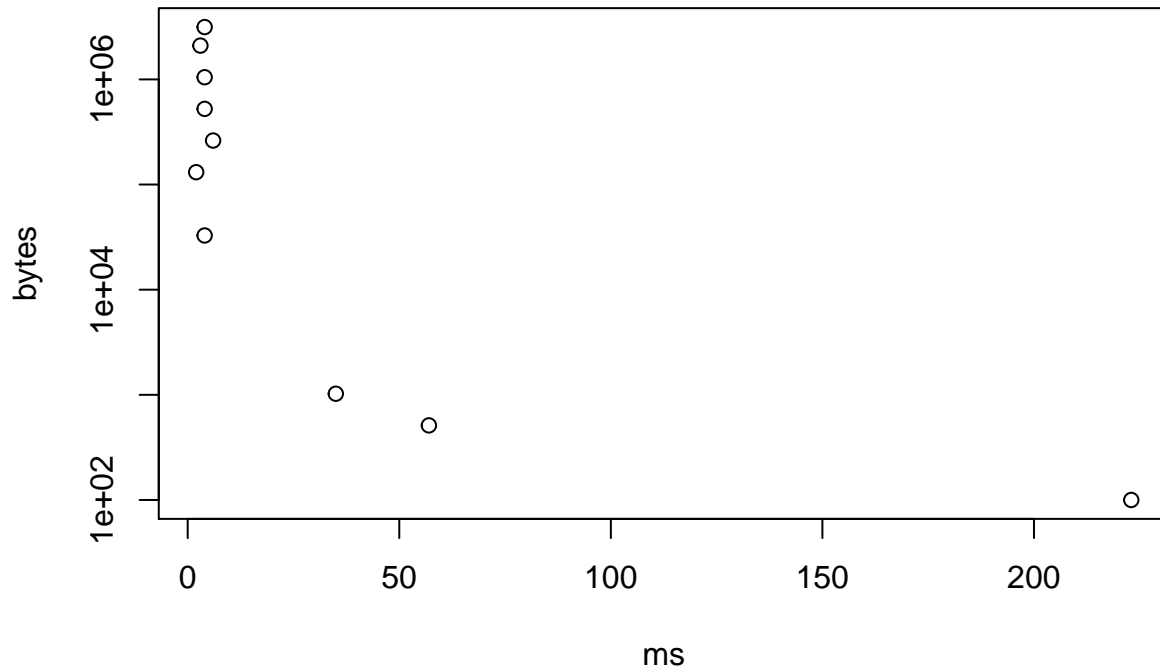
For the following plots the logarithmic scale was used for the number of bytes to clearly demonstrate the data rate.

The text file used to conduct the experiments was 100MB - 104857600 total bytes long. The following is a plot representation of the bytes written and the number of ms it took to write the whole file.

Block Sizes Used: [100B, 512B, 1KB, 32KB, 125KB, 256KB, 512KB, 1MB, 2MB, 3MB]

The following data was generated on my Mac machine:

Write data rate versus block size on Mac OS X



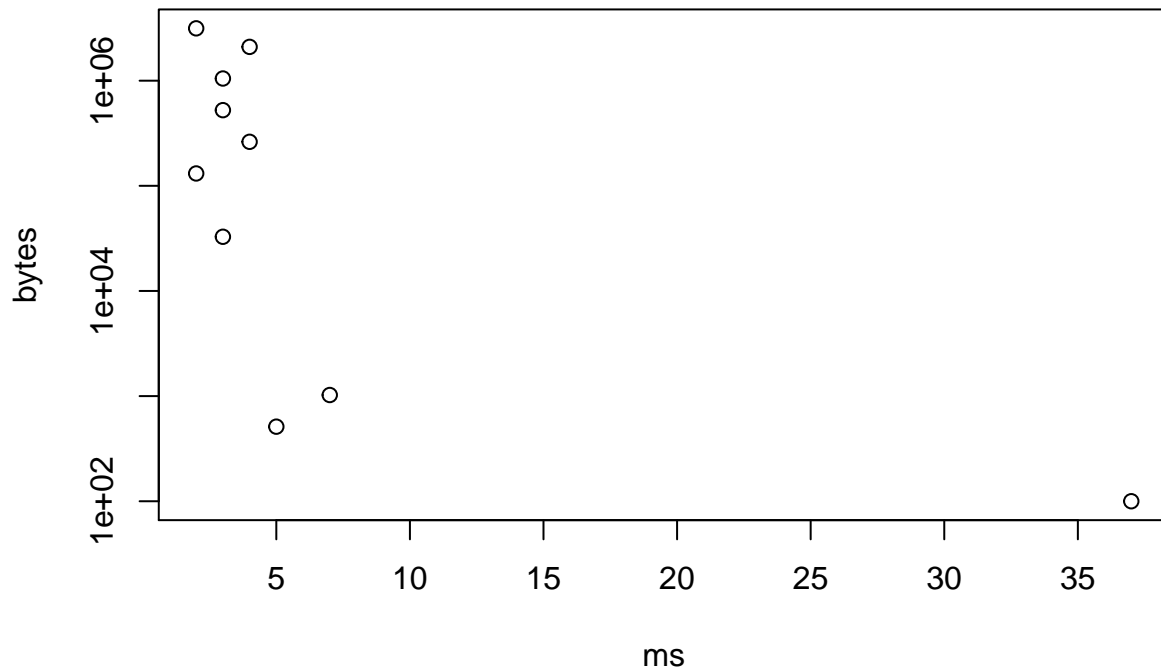
We can see that the most costly operation is distributing the 100MB of data into smaller sized blocks, with 100 bytes being the most costly taking 223ms, 512 bytes taking 57ms, 1024 bytes taking 35ms.

2. Optimal block size for write.

The most optimal block size appears to be 128KB only taking 2ms. And as the block size is increased up to and including 3MB, the number of ms it takes more or less plateaus at 4ms. It is probable that if the ratio of file size over block size decreases i.e. we continue to increase the block size beyond 3MB, the time it would take to write the file would increase as such a large size for a block size will become inefficient.

3. Comparison

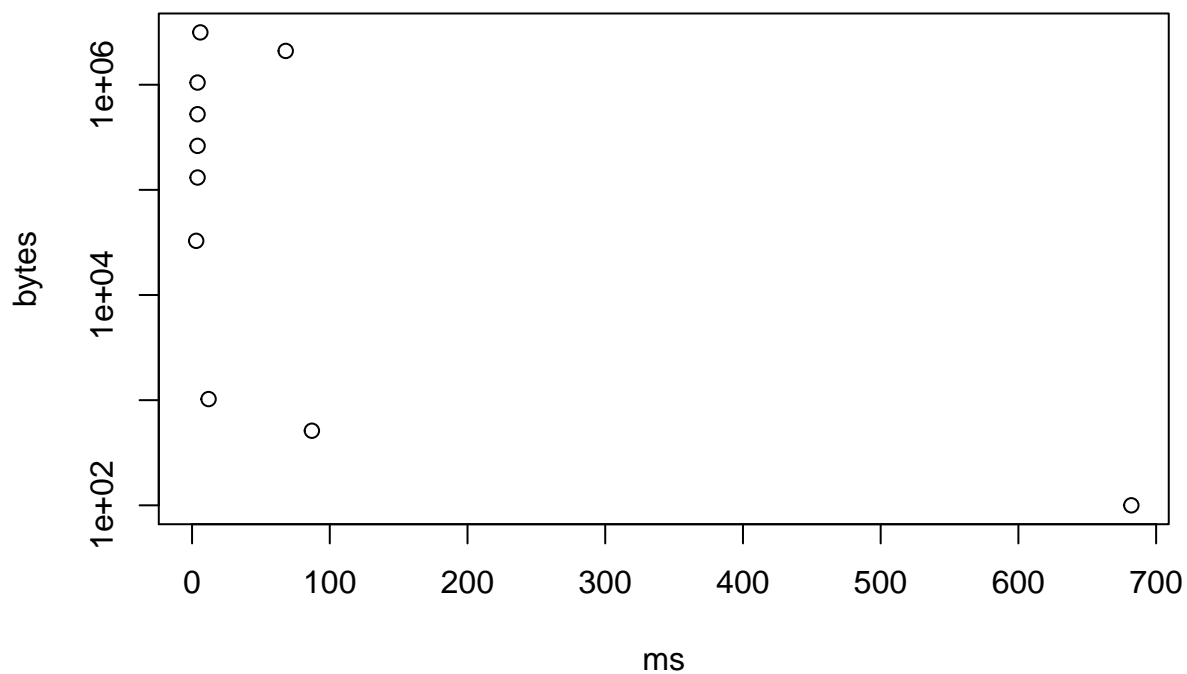
Write data rate versus block size on CDF



CDF machines turn out to be more efficient for writing data to blocks of a small size.

It took only 32ms to write 100MB of data for block size of 100B. Compare that with 223ms to perform the same operation on a Mac. However, the optimal block size is still 128KB. And there does not appear to be any difference in efficiency when it comes to block sizes greater than 128KB to 3MB as both machines perform similarly.

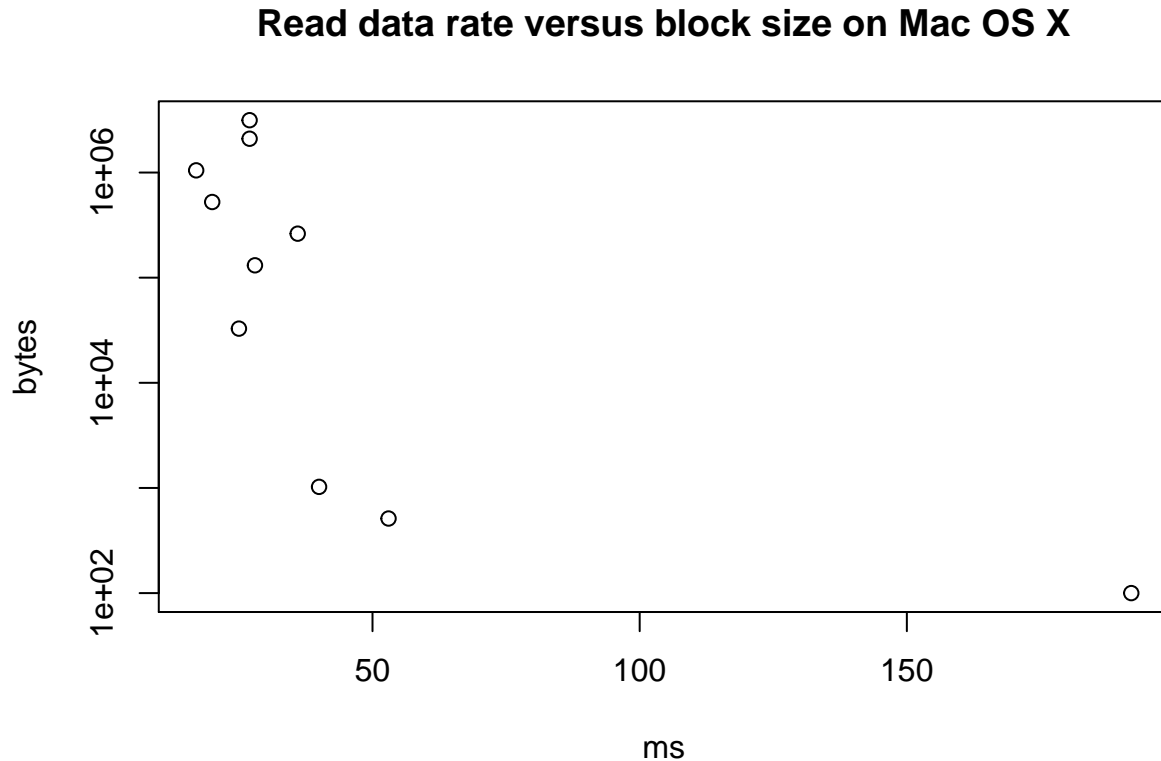
Write data rate versus block size on USB storage



Expectedly, writing the data to a USB storage is extremely inefficient if data is written in small block sizes (for a block size of 100B it took 682ms to write the whole 100MB file). However, like with the previous storage mediums, there was a sharp decrease in the amount of time it took as the block size increased up to 1MB. The optimal block size for USB storage is 32KB instead of 128KB and there was a sharp spike in the time it took at about 2MB. It appears to be the case that as the block size is increased, the time it will take to write the file will increase as well.

Sequential read from file

1. Plot of the observed read data rate versus block size.



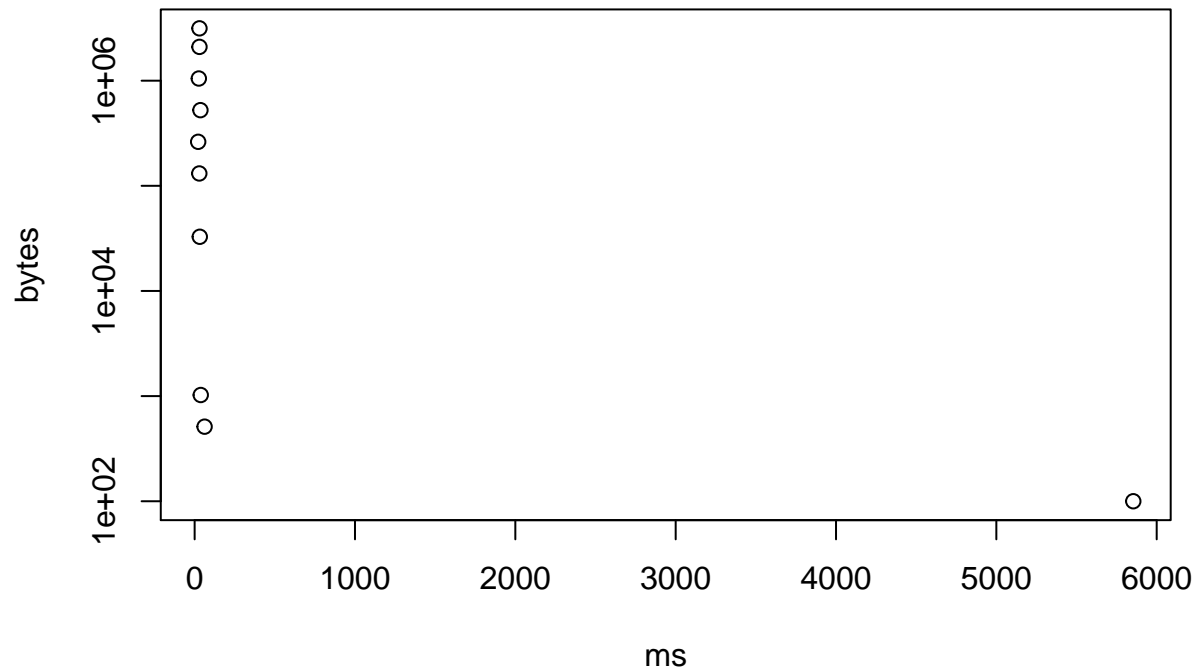
Similar to the write data vs block size ratio, the most costly operations involve the smaller block sizes with the block size of 100B taking 192ms to read the 100MB file. The overall trend appears to be a decrease in the time it takes to read the data as the block size is increased, however the decrease is not monotonic.

2. Optimal Block Size

Optimal read block, unlike the optimal write block is, 1MG. However, due to the oscillations in the number of ms it takes, this might not be a particularly accurate figure.

3. Comparison

Read data rate versus block size on USB storage



For this experiment, the file was stored on USB storage and read from the local machine. It was extremely costly to read the data for the first time in 100B blocks, however that might have something to do with accessing the USB stored file for the first time. The succeeding accesses are more in line with when the data is stored locally. Again, there's oscillation so the optimal block might not have been found, however, 256KB seems to be the optimal block size.