Binary File IO Test

Introduction

The purpose of this test is to evaluate the speed at which python can open, write to, and close files in binary format. This research is necessary to determine if a totally file based data storage method is viable for the Get Away Special Passive Attitude Control Satellite (GASPACS) mission. The results were somewhat mixed but likely still within acceptable parameters for the mission.

Methods

There were four processes tested in this procedure:

1. The time taken to open a binary file with Python.
2. The time taken to write a byte array of size 128 bytes to a binary file in Python.
3. The time taken to close a binary file with Python:
4. The time taken to open, write to, and close a binary file with the same byte array.

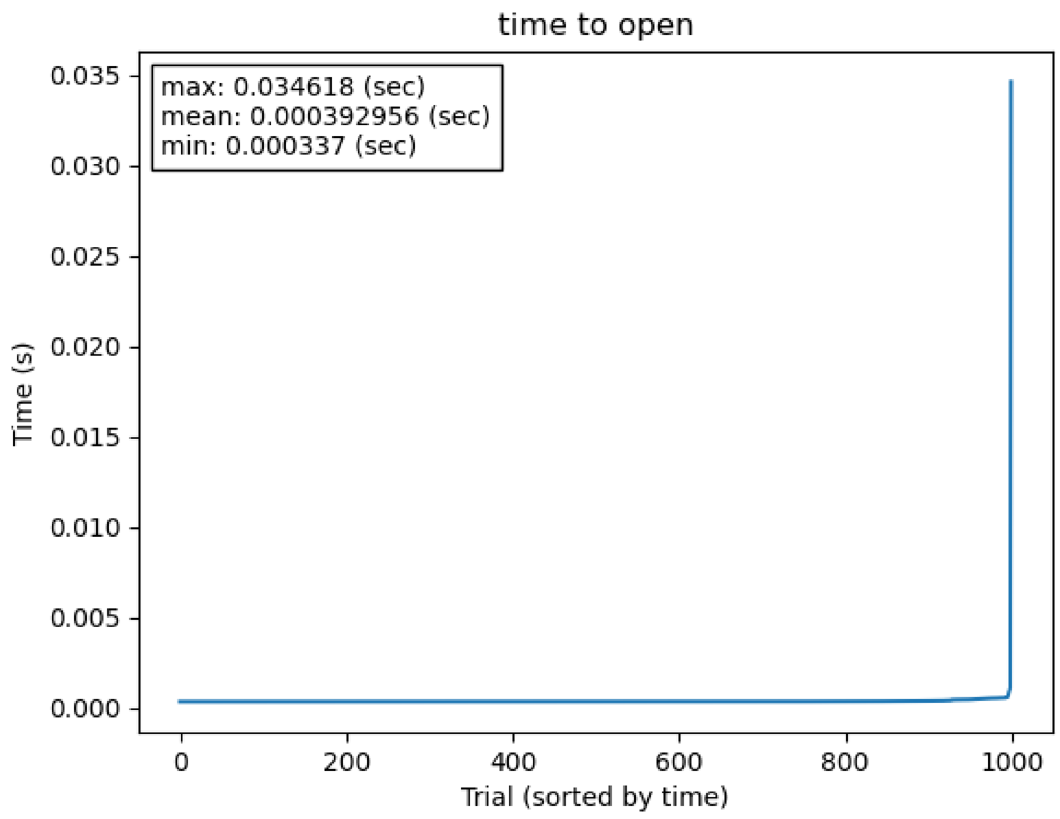
For each of the four trials, a 128 byte byte array was written to five files, 1000 times. At the end of the trial, each file contained 5000 identical byte arrays. Each byte array was a sequence of 127 "A" characters followed by one "B" character. An ascii character is exactly one byte so the packet size came out to be exactly 128 bytes. There should be no difference between writing a static byte array and writing random data to the file, as it takes the same amount of time to write a binary 0 to the file as it takes to write a binary 1. The trials were run on a Raspberry Pi Zero W. Each operation was timed individually in each trial resulting in a total of 20,000 data points. The plots in this paper, while not technically standard, illustrate well the proportion of the different speeds recorded while timing the operations. Each data point was sorted and plotted on a line graph. The thing to consider with these plots is the proportion of the low data points to the high data points.

Results

The test code was written such that the only thing needed to run the experiment was Python3 and Pip. The dependencies were installed by running `pip3 install -r requirements.txt` in the project directory, and then running `python3 main.py` to run the tests.

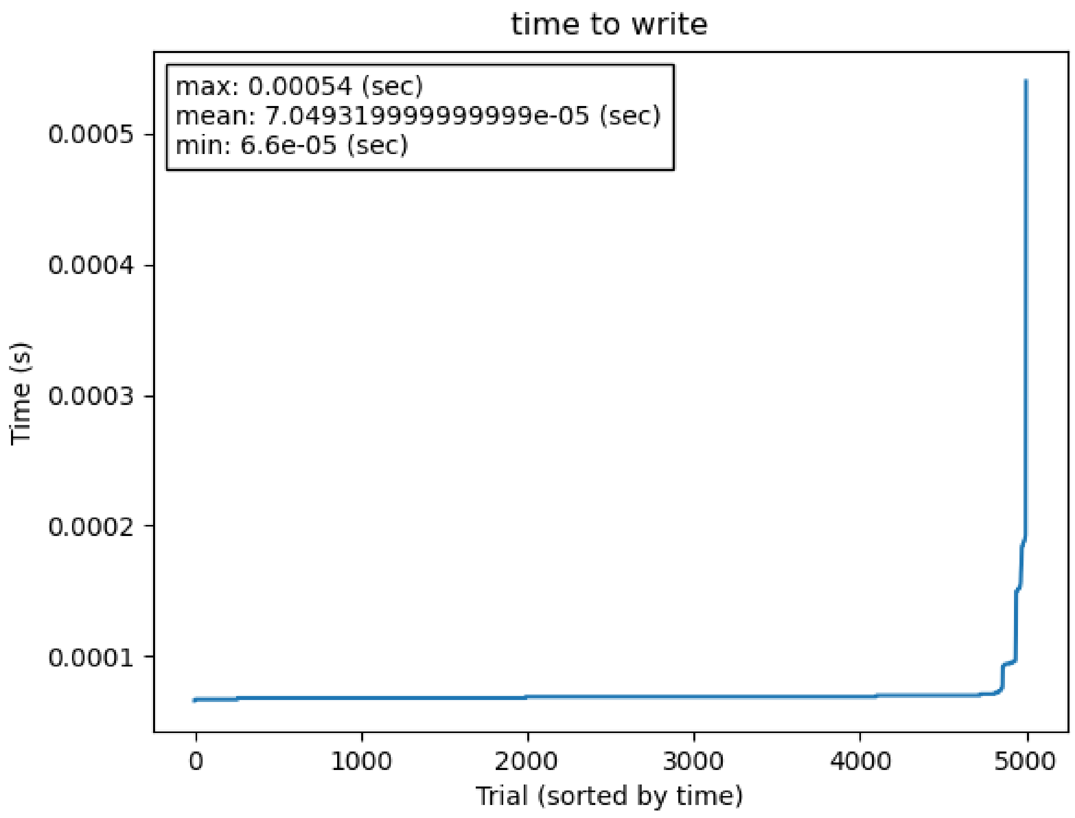
Trial One: Time to Open

The time taken to open the binary file was over all very miniscule with one or two outlying data points. The minimum time taken to open the file was 0.000337 seconds. The maximum time taken in this trial was 0.034618 seconds. This was a pretty large disparity, but only occurred in a hand full of open operations. To show how few, the average time taken between all 5000 operations was 0.000393 seconds as shown in the following figure.



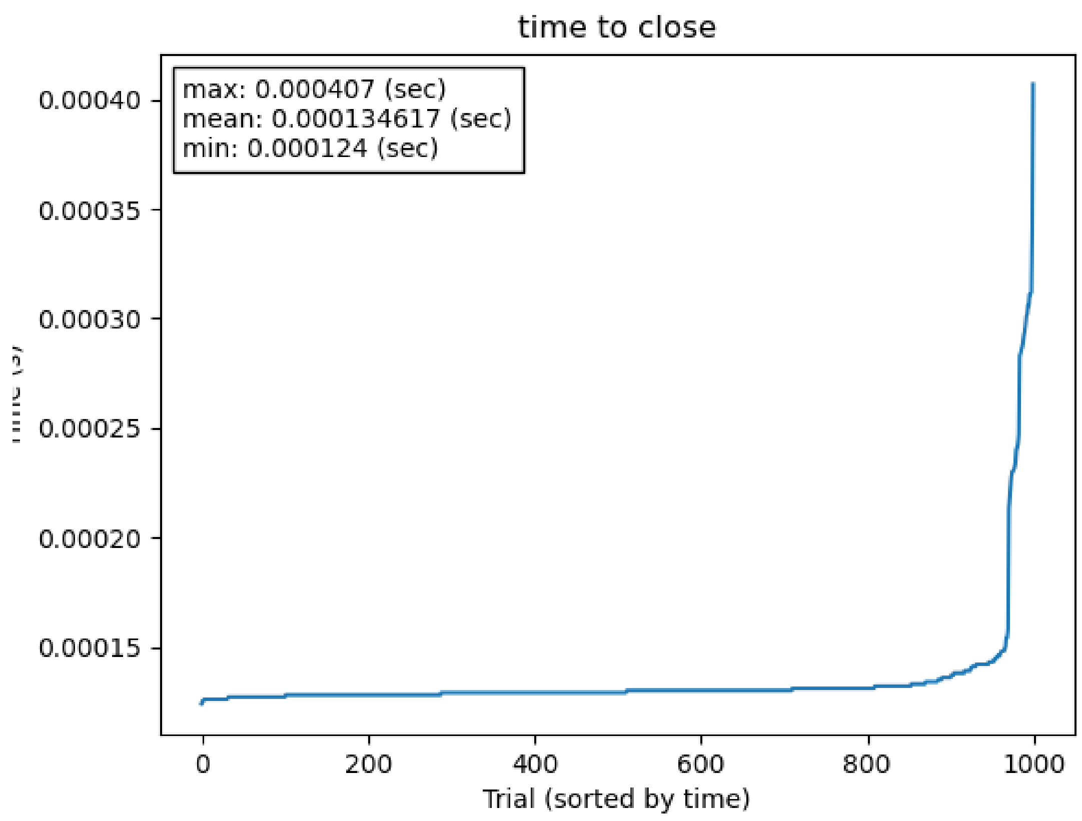
Trial Two: Time to Write

The write times for a byte array of size 128 bytes was miniscule. The minimum time taken was 6.6x10^-5 seconds or 66 microseconds. The maximum time for a write operation was 0.00054 seconds, and the mean operation time was 7.04x10^-5 seconds, or 70.4 microseconds as shown in the following figure.



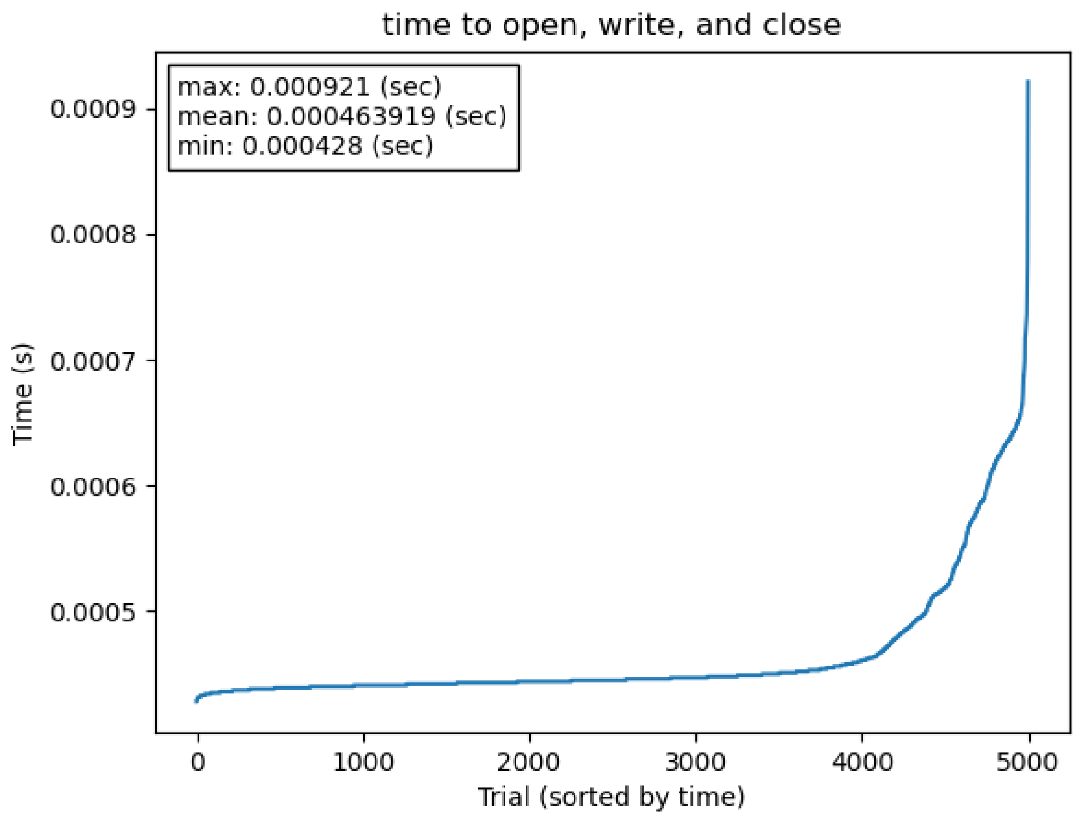
Trial Three: Time to Close

While slower than the write speed, the close speed was still sub millisecond. The minimum time was 0.000124 seconds and the maximum time was 0.000407 seconds. Judging by the figure below, the mean looks like it should be proportionately higher than the previous two trials, but the maximum was small enough that the mean was 0.000135 seconds, which was very close to the minimum time.



Trial Four: Open, Write, and Close

The combination of the previous three trials provided surprising times when compared specifically to the open time trial. All operations were timed at sub millisecond speeds. The minimum time was 0.000428 seconds, the maximum time was 0.000921 seconds, and the mean time was 0.000464 seconds. The plot below shows the results.



Conclusion

The important thing to remember when considering the viability of the binary file storage is enough time must be allowed for the worst case to occur. In terms of this trial, the longest operation took 0.035 seconds in the open trial. This may seem counter-intuitive, but that long operation could have easily occurred in trial four, which would have made the entire open, write, and close operation very slow. Now, given the distribution of the data, it's safe to assume that a long operation won't happen very often, but the fact that it *can* happen is important. If this behavior can be observed in a trial with as few as 5000 data points, it will certainly happen in production and must be prepared for.