Jeremiah Dixon

Software Engineering

Dr. Yi Zhou

November 11th, 2018

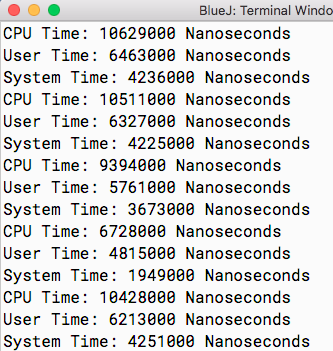
Project Report:

I/O Application Performance Evaluation

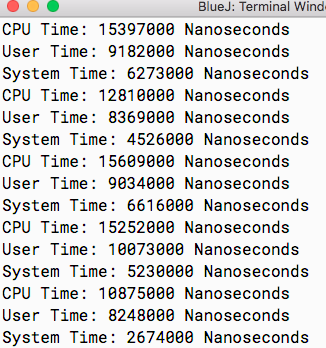
**Performance of I/O Application**

So what I did to conduct this experiment, I ran the I/O app 5 times each for 5 different file sizes for a total of 25 tests. I then gathered all the data times for each one respectively and took the mean for each to put into the graph. Here are the results for each one separately and then the graph to show the comparison of time to file data size.

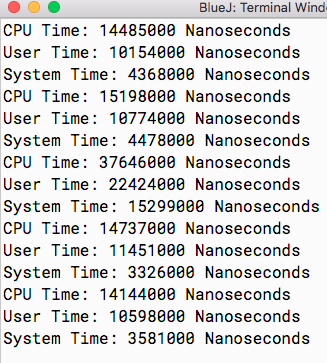
**100KB:** Average CPU Time: 9,538,000 Ns || Average I/O Time: 3,666,800 Ns

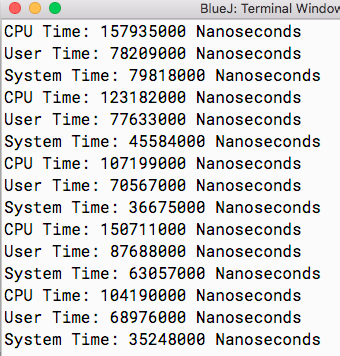
****

**500KB:** Average CPU Time: 13,988,600 Ns || Average I/O Time: 5,063,800 Ns

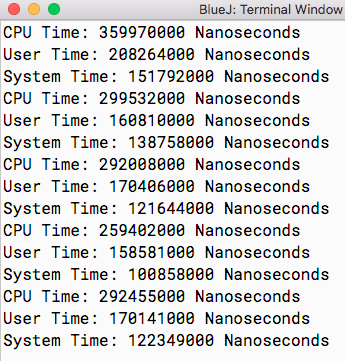
****

**1MB:** Average CPU Time: 19,242,000 Ns || Average I/O Time: 6,210,400 Ns

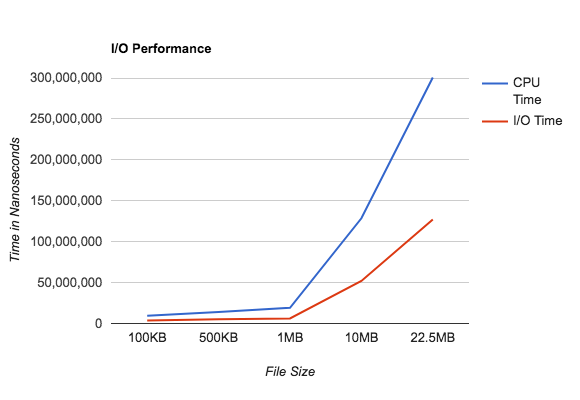
****

**10MB:** Average CPU Time: 128,634,400 Ns || Average I/O Time: 52,076,400 Ns****

**22.5MB:** Average CPU Time: 300,673,400 Ns || Average I/O Time: 127,080,200 Ns

****

**I/O Performance Graph**

****

**Performance Evaluation**

**Experimental Setup**:

Computer: MacBook Pro (13-inch, Mid 2012)

OS: macOS High Sierra ver. 10.13.6

Processor: 2.5 GHz intel Core i5

Memory: 4 GB 1600 MHz DDR3

Scripting Language: Java

IDE: BluJay

**Observations**: From the figure I am able to determine that the higher the data file the more time it takes for the I/O application. I am also able to see that the file size doubling doesn’t necessarily mean the I/O time or CPU time doubles i.e. from 500 KB to 1 MB the difference in time is not double the time of 500 KB. Also, the CPU time is always higher than the I/O time however they increase at a similar rate.

**Reasons**: The trend is obvious, the more data the I/O has to input and output the longer it takes. There may be a variance in performance because of the system I am using, or the other tasks/ threads/ processes/ jobs going on in my computer at the same time as the program. However the most obvious reason for a correlation between an increase in data size and increase in time taken is that the process must run for a longer time to input and output more data.

**Indications**: This indicates that increasing data size doesn’t necessarily decrease efficiency. In fact since doubling file size doesn’t double time taken the program is actually working more efficiently with more data.

**Performance Modeling and Validation**

**Simple I/O Performance Model:**

CPU TIME :

X = Data Size in KB, Y = CPU Time

Y = 6.0312 + 0.0129X

I/O TIME :

X = Data Size in KB, Y = I/O Time

Y = 1.2022 + 0.0055X

**Validation:**

I am plugging in file sizes that I have not tested into my simple prediction model to see the results and then comparing the results with my previously collected data and putting them into a graph together.

CPU TIME:

70.5 = 6.0132 + 0.0129(5000)

199.5 = 6.0132 + 0.0129(15000)

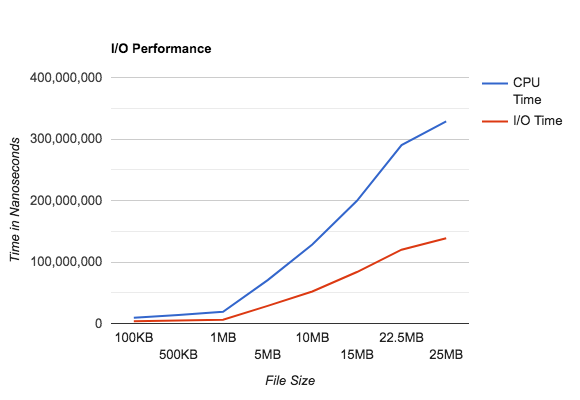
328.5 = 6.0132 + 0.0129(25000)

I/O TIME:

28.7 = 1.2022 + 0.0055(5000)

83.7 = 1.2022 + 0.0055(15000)

138.7 = 1.2022 + 0.0055(25000)



As you can see from the graph, the predicted CPU and I/O times fall right in line with the previous data collected. This model is accurate on estimating the times based in input data file size.

This is the system with 4 terabytes of data:

5.5405079275137 × 10^7 = 6.0132 + 0.0129(4294966919.53)

2.3622319259615 × 10^7 = 1.2022 + 0.0055(4294966919.53)

**Recommendation:**

If you will be using this system for huge levels of data, such as 100 terabytes or more then I highly recommend having some parallel system in place to help speed things up.