Dictionary Generation for a Markov Chain Text Generator

Fransandreimmanuel Colendres
Electrical and Electronics Engineering Institute
University of the Philippines, Diliman
Quezon City, Philippines
fransandreimmanuel.colendres@eee.upd.edu.ph

***Abstract*—This project explores the processing of large amounts of text-based data into a dictionary data structure that can be used for automated text generation that employs the concept of Markov chains. The project implements a program capable of taking text-based data, building a dictionary, and generating sentences using the information it derives from the input. The program was made to utilize a single process thread, but different versions of the program utilizing multiple process threads were created, to see if multi-threading can improve the runtime of the program. Using an input data sets containing as many as 1,218,252 words, it was found that the single-threaded version was significantly faster than the multi-threaded versions, regardless of input data size, and machine hardware. This could be attributed to sub-optimal multi-threaded programming, or the usecase being too simple to warrant any significant benefit from multi-threading.**

I. Introduction

With the ubiquity of technology and increasing connectivity, large amounts of data has become widely available to us. These voluminous sources of data have a potential to yield useful information when processed and analyzed correctly. The data available to us is so large, however, that it is no easy feat to process [1]. By creating a program that can learn to form its own sentences using an input text-based data, the project aims to be a small-scale emulation of a simple way large amounts of data can be analyzed. The program will use the statistical concept of Markov chains in order to achieve this. Using the concept of threading learned in CoE 135, the project also attempts to achieve a faster means of processing the input data by multi-threading the process. The expected outcome of this project is to have a multi-threaded program that is able to process the input texts faster than its single-threaded counterpart.

II. Review of Related Work

A. Markov Chains

In statistics, a stochastic process is defined as a sequence of variables where the value is determined by some probability. A Markov process is a stochastic process that has a finite number of states whose outcomes are only dependent on the previous states, and that its probabilities are constant with respect to time. A Markov chain is simply a Markov process that happens in discrete-time. A Markov chain is a set of states wherein the process starts in a certain state and moves from one state to another. For every unique state the process can be currently in, there are varying probabilities of what the next state can be. These probabilities are called transition probabilities, and they do not depend on the past states. For example, for a process that went from state A to B, the probability of the process going to state C from state B would be the same as if the process started in state D, then went to state B. For every Markov chain, the probability for a process to start on a specific state is defined with a vector of probabilities called the initial probability vector. Although Markov models only take into account the current state, the "current state" being considered can take the form of the previous n states. For example, a Markov model can be formulated where the probability to go to state D after going from state A, to state B, to state C is different to that of the probability to go to state D after going from state F, to G, to C. These Markov models are referred to as an n^{th}-order Markov model, where n is the number of previous states being considered. For instance, a 1^{st}-order Markov chain takes into account only the current state, whereas a 2^{nd}-order Markov chain takes into account the current state, and the state before the current state [2], [3].

B. N-gram Language Model

In tasks related to processing language, a language model is core to being able to analyze the data given. A common language model used is the N-gram language model. This language model allows for the derivation of the probability of the next word given the "history" of words. An N-gram language model operates under the Markov assumption, and thus, an N-gram language model is essentially an $(n-1)^{th}$- order Markov chain. For instance, a bigram language model, which determines the next word, given only the previous word, is a 1^{st}-order Markov chain [4].

III. Methodology

The program was implemented using the C programming language. A Markov chain text generator was implemented using the bigram language model.

A. Word Parsing

The program will process two consecutive words at a time: a "prefix" and a "suffix". Each word is stored in a string variable. For every new file, the process starts with both strings empty. The process then reads the text file one character at a time, and storing the read characters in the "suffix" string. Once the program reaches a blank space character, the "suffix" string is null-terminated, and the dictionary will be updated with the bigram parsed. Once the dictionary is updated, the "suffix" string is copied into the "prefix" string, and the program reads the next word, overwriting the "suffix".

B. Dictionary Building

The dictionary data structure was implemented using a two-layered hash table to store the "prefix" and its possible "suffixes" and another hash table to store the possible initial words that begins the generated text. The ""prefix and the "suffix" are both hashed using the murmur3 hash function using the murmur3 C function created by Peter Scott [5]. The first layer of the two-layered hash table, as well as the hash table containing the initial words contains 32768 entries. This value was determined by testing the utilization and collision rate of a varying number of table entries. Since the output of the murmur3 hash function is a 32-bit value, the number of entries would be determined by the lower n bits of the murmur3 hash output. Using a sample set of more than 200,000 unique english words, it was found that having hash table entries corresponding to the lower15 bits of the murmur3 hash had a good comprimise of entry utilization and number

of collisions. Each entry in the first layer of hash table of the two-layered hash table contains a "prefix", the number of its occurences, and a hash table of 16 entries. This smaller hash table contains the possible "suffixes" of the "prefix", and the number of occurences of the "suffix". Collisions are resolved using chained hashing. The dictionary is built by simply looking for the entry of the "prefix" being processed, updating its number of occurences, and updating the number of occurences of the "suffix" being processed, given the "prefix". If the "prefix" is an empty string, the "suffix" is stored in the hash table outlining the possible initial words of text generation.

C. Multi-threaded implementation

For this project, two versions of a multi-threaded implementation was created. The first method, henceforth referred to as "method 1", involves giving one thread a different text file each, and letting each thread process the entirety of the text file. This is repeated until all the text files are processed. The second method, which will be referred to as "method 2", makes each thread process a portion of all of the text files all threads will process roughly an equal portion of each text file. For instance, if there are four threads, each thread processes $\frac{1}{4}$ of each text file. Both methods use at least two mutex locks: one that restricts access to the two-layered hash table containing the "prefix" and its possible "suffixes", and another mutex lock for the hash table containing the different possible initial words for text generation. Method 1 uses an extra mutex lock that restricts access to the variable containing which text file is next to be processed. This essentially makes the threads "call dibs" on what file to process.

D. Text generation

Text generation is done after all the text files are processed. This part of the process cannot be multithreaded since it is a strictly sequential process where the next word to be generated relies on the previous word generated. Since the program keeps track of the number of occurences of each "prefix" and the number of occurences of each "suffix" given the prefix, the probability of each "suffix" being the word that appears after a given "prefix" can be computed by simply dividing the "suffix" occurence given the "prefix" by the number of "prefix" occurences. By traversing the possible "suffixes" of a prefix, a cumulative distributive function can be extrapolated by simply adding all the probabilities as the list of "suffixes" are traversed. To generate a word, a random number between 0 and 1 is generated, and a dummy variable is initialized to 1. Given a "prefix", the program traverses through all its possible "suffixes", subtracting the "suffix" probability from the dummy variable. Once the dummy variable is less than the random number generated, the "suffix" whose probability caused the dummy variable to become less than the random number is generated. The printed word then becomes the "prefix" to be considered next. A similar process is done to determine the first word to be generated.

E. Testing parameters

The performance of the programs were determined by measuring the time it took for the program to finish building the dictionary. The effect of hardware on the performance of the programs was tested by running the programs on two different machines. The first machine was a laptop with an i5 7200U processor, which has 2 physical cores and 2 threads per core, running at 2.5 GHz, and with 8 GB of RAM. The second machine was a virtual machine running on a desktop featuring an i5 4670k, which has 4 physical cores, and 2 threads per core. The virtual machine was allotted 4 GB of RAM, and 2 CPU cores. The scalability of the programs were also tested by varying the amount of data the programs needed to process. The data set sizes ranged from 1,660 words to 1,218,252 words. In addition, the effect of the number of threads being employed on the performance of the multi-threaded programs was observed by having the programs run using 2, 4, and 8 threads. Thirty runs were done for data sizes 101,821 words and 1,218,252 words for all versions of the program (single thread, methods 1 and 2 using 2, 4, and 8 threads). The input data consists of various written works obtained from the internet and stored in .txt files. The full list of texts used can be found in Appendix B. To save time, only the first 101,821 words come from unique written works. These 101,821 words of input was just duplicated to form a larger data set. Two-way ANOVA, one-way ANOVA and t-test statistical tests were done to compare the performances of the processes under the different testing parameters. The comparisons include performance of the program on different machines, the performance of the single threaded version to the multi-threaded versions, and the performance of the two multi-threaded methods.

IV. Results

A. Effect of hardware on the performance of the programs

Fig. 1 and Fig.2 graphs the time it takes the programs to process the given input data on the laptop setup, while Fig. 3 and Fig.4 graphs the same information, but for the desktop setup. Appendix A contains all of the results of testing for those interested in the numerical data collected. At a glance, it seems that the programs perform better on the virtual machine setup compared to the laptop setup. To confirm, a two-way ANOVA statistical analysis was conducted using the data gathered from the 30 trials of processing 1,218,252 words comparing both setups, and the single-threaded, and both multi-threaded methods using 4 threads. The null hypotheses used were that there is no significant difference in processing time for both laptop and virtual machine; that there is no significant difference in processing time for the single-threaded and multi-threaded methods; and that there is no interaction between these two independent variables. It was found that the p-score relating the interaction of the independent variables was less than the decided p-value of 0.05, meaning that there is significant interaction between the two independent variables. To confirm if there is a difference in processing time due to hardware, three t-test statistical analyses were done comparing the performances of the laptop and virtual machine setups when it comes to running the single-threaded program, the multi-threaded program using method 1 on 4 threads, and the multi-threaded program using method 2 on 4 threads. These tests were also done with a 95% confidence, and using the results of processing 1,218,252 words. After the three t-tests, it was found that, for all three tests, there was a significant difference in processing time due to hardware. It can be said that hardware affects the processing time of the programs created. Since the programs have a faster processing time on the virtual machine setup, subsequent trials to gather data was done on the virtual machine setup. The higher clock speed of the CPU on the virtual machine set up was beneficial to the programs since a higher CPU clock speed means that the CPU can accomplish tasks faster. A stronger CPU will also benefit multi-threading. As seen in Fig. 1 and Fig.2, the laptop's multi-threaded performance on 2 threads was faster than those on 4 and 8 threads. This may be due to the fact that the laptop has a weaker CPU, and thus, the additional context switching brought about by more threads causes a slower execution time since the CPU is too weak to handle the additional threads in a timely manner. Compare this to the multi-threaded performances of the virtual

machine seen on Fig. 3 and Fig.4, where program run time was faster on 4 and 8 threads.

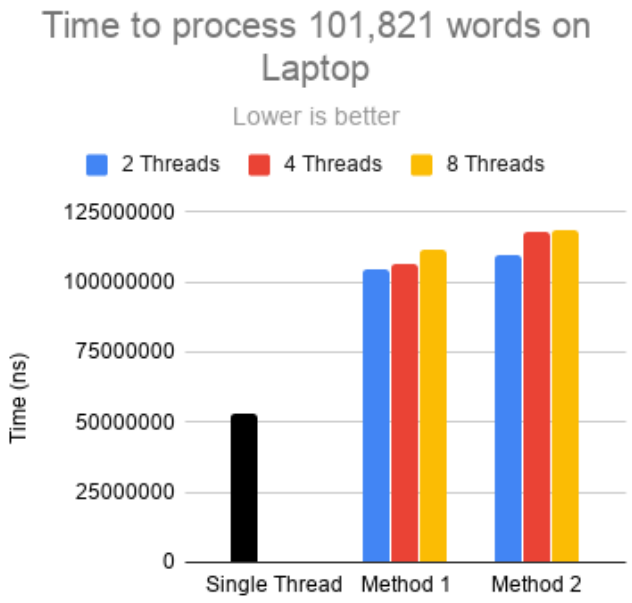

Fig. 1. Time it takes to process 101,821 words on laptop setup

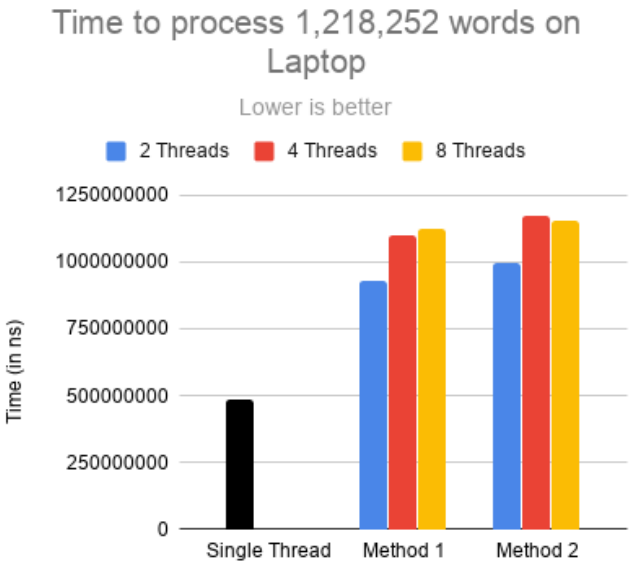

Fig. 2. Time it takes to process 1,218,252 words on laptop setup

B. Comparison of single-threaded performance and multi-threaded performance

To find out if there is a significant difference in processing time depending on the version of the program used, a one-way ANOVA statistical test was conducted, comparing the performances of the three versions of the program. For this analysis, we used

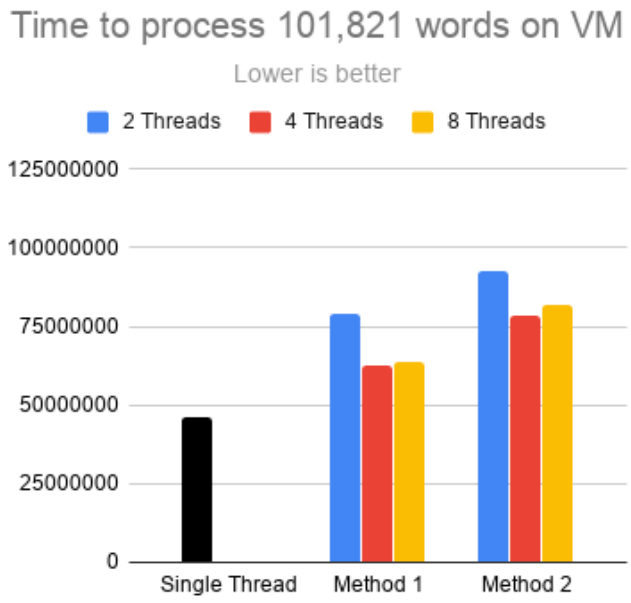

Fig. 3. Time it takes to process 101,821 words on laptop setup

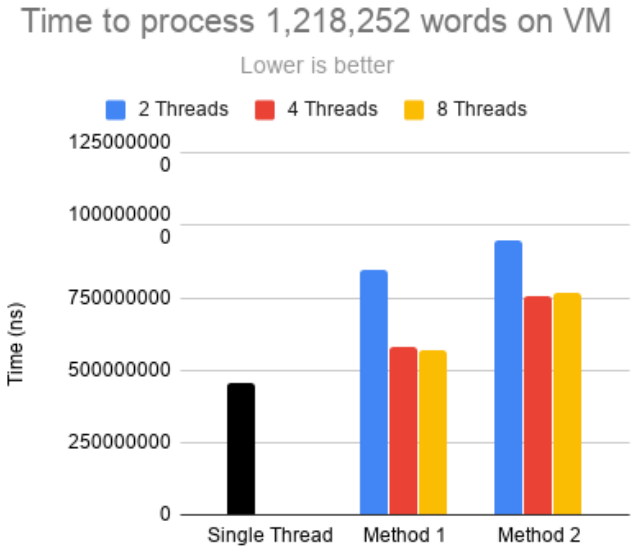

Fig. 4. Time it takes to process 1,218,252 words on laptop setup

the processing times of the single-threaded, and both multi-threaded methods using 4 threads. The one-way ANOVA was done twice: once to compare the results of processing 101,821 words and another to compare the results of processing 1,218,252 words. This is to see how well the programs scale with respect to one another. Once again using a 95% confidence interval, and with a null hypothesis stating that the average processing time is the same for all methods, the resulting p-scores resulted in the rejection of the null hypotheses. It can be said that, for both data set sizes used in testing, the single threaded version of the program is significantly faster than the multi-threaded versions. The slower multi-threaded versions may have been

due to sub-optimal multi-threaded programming. The mutexes applied might have caused a significant overhead in the processing time, since all threads shared only two data structures for storage, both of which were safeguarded by a mutex lock. In addition, method 1 also had another mutex which may have caused further slow-downs. It is possible that better algorithms and more efficient mutex schemes may alleviate the slower processing of the multi-threaded methods. It might also be possible that the operations taken to process the data may have been too simple to actually benefit from multi-threading, since, in essence, the process just stores a word in a hash table and increments a number.

C. Comparison between the two multi-threaded versions of the program

To see which multi-threaded version is faster, a two-way ANOVA statistical test was done with 95% confidence interval. This two-way ANOVA compared the processing time of 1,218,252 words, and had the number of threads, and the multi-threaded method as the independent variables. The null hypotheses used for this test are that the average processing time is the same regardless of multi-threaded method; the average processing time is the same regardless of the number of threads used; and that there is no interaction between the independent variables. The results of the statistical test indicated that there is a significant interaction between the two independent variables, so no conclusions can be made just yet. To isolate the interaction of the independent variables, a one-way ANOVA was done twice. Each ANOVA test was to compare the performance of each method to itself, with the number of threads as the independent variable. It was found that, for both methods, there was a significant difference in processing time depending on the number of threads. Like mentioned before, the number of threads may have bearing on the performance depending on the type of hardware used. For slower machines, more threads could be detrimental since the CPU is not fast enough to accomodate the context switching in a timely manner. For faster machines, using fewer threads may not be using the processing core to its full potential. In addition, the concurrent processing that faster machines allow may offset any slowdowns due to context switching and waiting for mutex lock access.

To see if there was a difference in processing time depending on the multi-threaded method used, a t-test was done to compare both methods using 4 threads. This t-test was done twice: once on the results of processing 101,821 words, and another on the results of processing 1,218,252 words. This is to give a sense of how well the two versions scale with data size, with respect to one another. Using a 95% confidence interval, and a null hypothesis of the average processing time is the same regardless of method, it was found that there was a significant difference in processing time for both data set sizes. Given the graphical data, it can be said that method 1 is faster than method 2. This may be due to the fact that method 2 requires additional operations in order to determine how to divide a single text file among all the threads.

D. Scaling of Program with regards to data set size

To give a rough idea of how the program run time scales with the amount of data it needs to process, additional test runs of every method was done with varying data set sizes. In addition to the 30 trials of processing 101,821 and 1,218,252 words, 10 trials per method was also done to process 1,660, 10,622, 203,642, 509,105, 712,747, and 1,018,210 words. The averages of each trial was taken and plotted on Fig. 5 to give a visual sense of how the single threaded version scales with both multi-threaded methods using 4 threads. Fig. 6 graphs how the performance of method 1 scales depending on the number of threads used, with the single-threaded version as a benchmark. Fig. 7 is a similar graph as Fig. 6 but it represents method 2, instead.

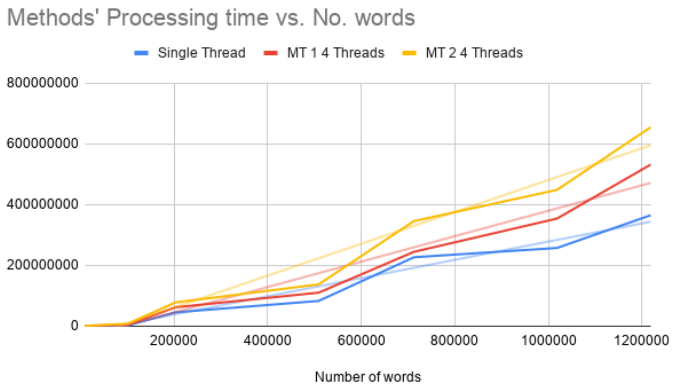

Fig. 5. Single-threaded vs. multi-threaded method 1 (4 threads) vs. multi-threaded method 2 (4 threads)

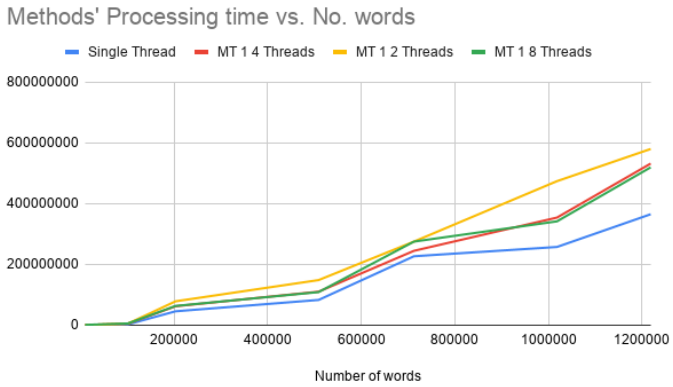

Fig. 6. Scaling of method 1 using varying number of threads

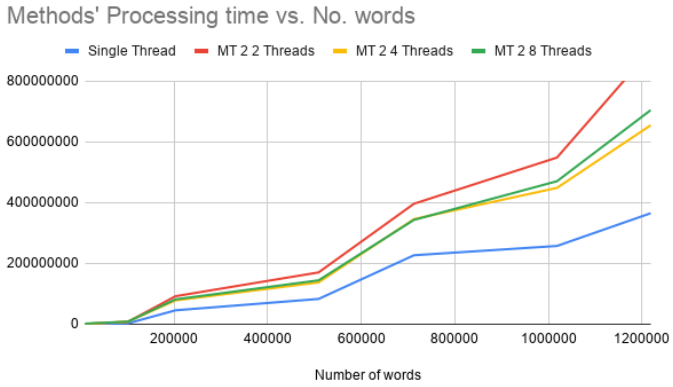

Fig. 7. Scaling of method 2 using varying number of threads

It can be seen that the single-threaded version is always the fastest, regardless of data size. For the multi-threaded methods, the

performance of using 4 threads is similar to the performance of using 8 threads, with 2 threads being the slowest, at least for this machine. In any case, it can be said that all programs scale linearly, but the slope of the multi-threaded methods are steeper than the single-threaded version. From the graphs obtained, the trend seems to indicate that the single-threaded method will remain the fastest even with larger data sizes.

E. Text generated

The text generated by the program was similar to that of the previously-made simple Markov chain text generators, or keyboard predictive text results. That is to say that the sentences, though structurally adhering to the rules of the english language, mostly end up sounding nonsensical. Every now and then, there is a result that makes sense, but most of the time it gives something resembling very abstract poetry, or something that almost makes sense, at best. An example output is given below:

> ""This is enslaved, all memory cube, tilted to be," Old Ones. No answer the explosion took from every second cycle. That was destined to conceal; but infinitely distant, but the goblin, "don't look next to sing with himself, and the even though apparently very inapt in the almighty."

At times, there are phrases identifiable to its source material. This is due to the fact that the words, or string of words, are unique to one sample text only, so when the program happens to decide to print the first word in the sequence, it is statistically impossible, or improbable, not to be followed by the same words as in the sample text. As an example, the program can sometimes print out "Cogito ergo sum". Since the word "Cogito" starts a new line in one of the sample texts, the program sometimes prints the word out to start the text generation chain. When this happens, "ergo" is sure to be printed next, and then "sum", since all of the texts used as input only ever had "ergo" following "Cogito" and "sum" following "ergo".

V. Conclusion

The programs implemented did not perform in the way they were expected to. More specifically, the multi-threaded programs were always slower than the single-threaded program. This could be due to sub-optimal multi-threaded programming, or having an operation that was too simple to benefit from multi-threading. In addition, it was found that for this set of programs implemented, hardware had an effect on the programs in-line with the lessons in CoE 135 regarding multi-threaded performance and speed of hardware. The same can also be said when it comes to the number of threads being employed. Multi-threaded method 1 was substantially faster than multi-threaded method 2, due to not needing to delimit the portion within a text file to be processed, unlike in method 2. In all versions of the program, the time to process data scaled linearly with the data size, with the multi-threaded versions having a steeper slope than that of the single-threaded version. This indicates that the single-threaded version will remain the fastest regardless of data size.

Appendix A
Raw performance data from the test runs

The tables from Table I to Table XXX tabulate the time it takes for the program to process the set amount of data, using the specified method.

TABLE I
TIME TO PROCESS 1,218,252 WORDS USING SINGLE-THREADED PROGRAM ON LAPTOP SETUP

		Time to run (ns)
Trial	1	478648036
	2	476024748
	3	482190999
	4	480987472
	5	505724914
	6	477713506
	7	494367646
	8	464280283
	9	470800370
	10	491778470
	11	458813931
	12	498323262
	13	497183277
	14	462694042
	15	480512834
	16	478230737
	17	474267253
	18	485563136
	19	489932387
	20	474637354
	21	461757505
	22	478572381
	23	521125996
	24	524406518
	25	465182764
	26	522507182
	27	520146153
	28	476494126
	29	479883657
	30	484178690
Mean		485230987.6
Std. Dev		18370127.13
Confidence Interval		6859518.205

Appendix B
List of texts used as program inputs

Table XXXI contains the list of all of the literary works used as input data for the program, its author, and its word count. A total of 33 unique texts were used in this project. As a time-saving measure, the texts were duplicated in order to reach the specified input word count sizes.

References

[1] What is big data? Online article. Oracle. [Online]. Available: https://www.oracle.com/ph/big-data/guide/what-is-big-data.html
[2] C. Grinstead and J. Snell, *Introduction to Probability*. American Mathematical Society, 1997, ch. 11, pp. 405–408.
[3] W. Hussen, "Markov processes," Lecture handout, Ohio State University. [Online]. Available: https://people.math.osu.edu/husen.1/teaching/571/markov_1.pdf
[4] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Prentice Hall, 1997, ch. 3, pp. 31–33.
[5] P. Scott. Murmur3 hash in c. C library. [Online]. Available: https://github.com/PeterScott/murmur3
[6] M. L. K. Jr., "I have a dream..." Speech.
[7] J. Swift, "A modest proposal," Essay.
[8] E. A. Poe, "The cask of amontillado," Short Story.
[9] ——, "A tell-tale heart," Short Story.
[10] A. Weir, "The egg," Short Story.
[11] I. Asimov, "The last question," Short Story.
[12] H. Ellison, "I have no mouth and i must scream," Short Story.
[13] U. K. L. Guin, "The ones who walk away from the omelas," Short Story.
[14] A. C. Clarke, "The nine billion names of god," Short Story.
[15] ——, "The star," Short Story.
[16] S. Jackson, "The lottery," Short Story.

TABLE II
TIME TO PROCESS 1,218,252 WORDS USING MULTI-THREADED PROGRAM (METHOD 1) ON LAPTOP SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trial	1	1091875596	915195372	1203556832
	2	1105353089	898048157	1127601544
	3	1092591381	931336903	1122106161
	4	1090779852	928186814	1104110777
	5	1107715039	917214973	1108863453
	6	1104878476	1010092615	1128033056
	7	1110117111	891698736	1174048202
	8	1098611528	944922152	1101746342
	9	1097215308	898803946	1116069583
	10	1127154792	919160558	1132549192
	11	1092162256	901456658	1118268057
	12	1097309701	915840820	1124795803
	13	1095915653	964275196	1109099170
	14	1038250073	950470953	1115870856
	15	1087655410	908255872	1104880638
	16	1105283762	933845824	1113538371
	17	1104442242	934331821	1133676320
	18	1125535588	932860956	1114457162
	19	1111749850	911105159	1118562665
	20	1116721268	901817585	1097437038
	21	1099358651	995424066	1119465860
	22	1088866218	937932552	1108076861
	23	1096543372	908109280	1106614588
	24	1101526524	903084942	1108773844
	25	1106980230	897102963	1118745770
	26	1133166474	995140665	1107284688
	27	1092838658	975216157	1113319285
	28	1095157356	883403557	1128574759
	29	1092493756	979872086	1111838053
	30	1105073657	878906692	1106586388
Mean		1100444096	928770467.7	1119951711
Std. Dev		16372566.66	34749930.45	21212379.84
Confidence Interval		6113616.865	12975837.28	7920832.807

TABLE III
TIME TO PROCESS 1,218,252 WORDS USING MULTI-THREADED PROGRAM (METHOD 2) ON LAPTOP SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trial	1	1165974672	985014657	1148225286
	2	1148119310	1045998378	1127044020
	3	1164445005	1032236847	1141816850
	4	1173329672	975751827	1162331691
	5	1189727919	1023614815	1143867322
	6	1158605302	1026700318	1141740243
	7	1160916935	971525641	1136438829
	8	1154980650	987376043	1185918032
	9	1156742239	966548074	1164343931
	10	1179971734	971877284	1144436450
	11	1163579011	980643339	1150119727
	12	1158736577	951700211	1169405130
	13	1162466532	1179492856	1152577100
	14	1160180000	972707940	1189511094
	15	1147040285	969126389	1135402058
	16	1157342737	952634575	1151214511
	17	1171512918	1006135374	1149125246
	18	1154287102	979655379	1147637705
	19	1184897885	956374558	1134023801
	20	1172722203	969720624	1190333717
	21	1169472095	1006994714	1146801208
	22	1179564013	992646516	1151642438
	23	1246299910	976728899	1147655086
	24	1207987001	983488213	1160169691
	25	1168367450	1021746919	1141144999
	26	1187136045	983925975	1135971807
	27	1183327907	991706143	1149000677
	28	1149811210	941492468	1152900104
	29	1178514785	1017427925	1151311812
	30	1157048631	1030580833	1144772777
Mean		1170436925	995052457.8	1151562778
Std. Dev		19999038.18	43904889.01	15508900.69
Confidence Interval		7467763.584	16394354.99	5791118.693

[17] H. Murakami, "On seeing the 100% beautiful girl one beautiful april morning," Short Story.
[18] "Disappointing: Microsoft confirmed that project scarlett is actually just a brothel they're building in thailand where xboxes can go to be pleasured," Satirical news article, The Onion.
[19] "'pokémon' fans are up in arms with game freak's refusal to include a national sex offender registry in 'sword and shield'," Satirical news article, The Onion.
[20] "Making amends: Blizzard added a drawing of xi jinping getting pinched on the ass by a crab to all spawn rooms on overwatch's lijiang tower map," Satirical news articles, The Onion.
[21] J. F. Kennedy, "Ich bin ein berliner speech," Speech.
[22] F. D. Roosevelt, "Infamy speech," Speech.
[23] I. Asimov, "Nightfall," Short Story.
[24] A. Bleyaert, "How to lose weight in 4 easy steps," Short Story.
[25] L. M. Alcott, "Cousin tribulation's story," Short Story.
[26] K. Chopin, "The story of an hour," Short Story.
[27] "Ali baba and the forty thieves," Short Story.
[28] H. P. Lovecraft, "The call of cthulhu," Short Story.
[29] W. W. Jacobs, "The monkey's paw," Short Story.
[30] H. C. Andersen, "The brave tin soldier," Short Story.
[31] N. Hawthorne, "The haunted mind," Short Story.
[32] J. Joyce, "Araby," Short Story.
[33] K. Chopin, "Regret," Short Story.
[34] O. Henry, "The gift of the magi," Short Story.
[35] H. P. Lovecraft, "Ex oblivione," Short Story.
[36] M. R. Rinehart, "The game," Short Story.
[37] L. Andreyev, "On the day of the crucifixion," Short Story.
[38] G. de Maupassant, "A dead woman's secret," Short Story.

TABLE IV
TIME TO PROCESS 101,821 WORDS USING SINGLE-THREADED PROGRAM ON LAPTOP SETUP

		Time to run (ns)
Trials	1	46858176
	2	47942347
	3	63563055
	4	50304424
	5	47245946
	6	47427802
	7	57396068
	8	62796096
	9	47465401
	10	49122095
	11	46162786
	12	61949859
	13	47858031
	14	48575142
	15	47517891
	16	48999671
	17	58305043
	18	48720141
	19	61349375
	20	48363234
	21	58514111
	22	63441071
	23	45986740
	24	48034349
	25	60781534
	26	48033138
	27	47826134
	28	47156153
	29	60241274
	30	64144911
Mean		52736066.6
Std. Dev		6689725.117
Confidence Interval		2497984.412

TABLE V
TIME TO PROCESS 101,821 WORDS USING MULTI-THREADED PROGRAM (METHOD 1) ON LAPTOP SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	99453287	103259043	98986756
	2	102054156	81752571	105646775
	3	104092676	134396443	122635456
	4	118567457	86006152	99738223
	5	114742385	110220605	119131020
	6	114737754	110066825	107998278
	7	99911703	107776482	120554956
	8	103212391	127715843	107410209
	9	99936064	96489122	121250687
	10	103346639	118492128	105924440
	11	101927613	136452935	108809011
	12	101610350	100512915	107435797
	13	115895392	93329658	113267513
	14	114628364	93618367	121719335
	15	118920149	96526769	105178606
	16	102318049	110568568	115719004
	17	112652893	94870072	106847928
	18	101473058	113380118	106085519
	19	102925153	92512405	105389259
	20	102137125	114772380	117710198
	21	118116700	97939863	106247485
	22	100364825	84484863	118688400
	23	103145602	136480196	105715102
	24	101564605	83724864	121947186
	25	102812839	88290040	108016395
	26	102812839	82472310	105384453
	27	116799391	85278454	119022028
	28	121581797	97478957	106065288
	29	100362969	116226500	122007494
	30	98227994	133938153	106074276
Mean		106677740.6	104301120	111220235.9
Std. Dev		7436627.198	17117372.02	7333250.755
Confidence Interval		2776882.232	6391731.756	2738280.834

TABLE VI
TIME TO PROCESS 101,821 WORDS USING MULTI-THREADED PROGRAM (METHOD 2) ON LAPTOP SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	127490306	130769992	113548199
	2	113236320	121909529	112909037
	3	118283484	92725756	116897527
	4	112636064	95433231	110611953
	5	111345627	139630989	126877372
	6	114817509	89174538	113671270
	7	112990593	91467335	111689579
	8	126224104	108445892	112806132
	9	111662466	91819012	113220258
	10	131557646	135207455	128611273
	11	113242295	143224292	121628167
	12	128775888	126876727	114305409
	13	111942477	90483814	128355287
	14	113495822	97752194	113510392
	15	112542400	120759083	117631308
	16	114535453	123621902	134640454
	17	111473543	138598778	126880799
	18	111536888	98576182	113131666
	19	123891152	100260539	113989379
	20	114935685	104608005	126413896
	21	126194756	107251097	114442536
	22	131221129	118516232	113702056
	23	111975172	93422845	132266612
	24	127383353	93354136	125770427
	25	112897409	93347241	114242989
	26	130396602	97019935	110778918
	27	113733988	90008569	114776798
	28	112747421	102128130	114984527
	29	114515888	118542622	124682281
	30	111649814	133731853	115925671
Mean		117644375.1	109622263.5	118430072.4
Std. Dev		7237783.426	17783997.79	7061810.877
Confidence Interval		2702632.748	6640653.91	2636923.518

TABLE VII
TIME TO PROCESS 1,218,252 WORDS USING SINGLE-THREADED PROGRAM ON VIRTUAL MACHINE SETUP

		Time to run (ns)
Trials	1	439708227
	2	436603156
	3	454928684
	4	435648400
	5	434853707
	6	441361115
	7	441997910
	8	421970141
	9	469442607
	10	469634578
	11	529201574
	12	432983070
	13	453889980
	14	447242539
	15	456773918
	16	448010432
	17	526891209
	18	490609937
	19	484510990
	20	451304563
	21	445654673
	22	443181162
	23	478653421
	24	433579598
	25	460233140
	26	460040204
	27	428557378
	28	442458450
	29	423157661
	30	433180199
Mean		453875420.8
Std. Dev		26426982.09
Confidence Interval		9867997.289

TABLE VIII
TIME TO PROCESS 1,218,252 WORDS USING MULTI-THREADED PROGRAM (METHOD 1) ON VIRTUAL MACHINE SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	552588777	863159732	544410474
	2	552750999	874830878	565597407
	3	559799138	894858526	574244563
	4	727491528	881789792	554530268
	5	562487507	862599577	565123801
	6	565247573	914429850	568477453
	7	556220674	844852848	545010776
	8	553751446	883814712	563329192
	9	557065109	878063582	614315800
	10	566577586	884595112	562856851
	11	549910614	880151705	545575106
	12	550130238	887693935	573631408
	13	561222935	876278348	555859758
	14	567152798	879342643	550309750
	15	550379728	891760604	568878361
	16	644127920	891054833	551410206
	17	682048789	838946026	602179263
	18	626450441	800774543	571667046
	19	697007525	816480198	583048714
	20	586479243	803132648	556802624
	21	567823993	824173815	587728329
	22	563352575	818769213	579742504
	23	544528938	812052590	578652658
	24	555091648	798174606	590458681
	25	598308687	807326337	586860618
	26	564973751	789633267	558202671
	27	555092591	798134315	563726823
	28	548884608	808823805	571225199
	29	573410475	803876665	551443814
	30	558873192	793902604	548492490
Mean		579974367.5	846782577	567793086.9
Std. Dev		47231359.53	39407673.45	17216894.67
Confidence Interval		17636479.5	14715067.1	6428894.127

TABLE IX
TIME TO PROCESS 1,218,252 WORDS USING MULTI-THREADED PROGRAM (METHOD 2) ON VIRTUAL MACHINE SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	742180953	937153784	757626774
	2	739452522	949810759	769634016
	3	720276523	965284628	747376566
	4	737294334	922387910	775191610
	5	758490605	934975389	746661916
	6	749678003	959561957	768369902
	7	731217006	964597956	747697200
	8	731342967	938591563	749581350
	9	724522793	945111956	823255015
	10	766734925	936532361	781286651
	11	732857317	986375214	790548355
	12	757431762	1000868209	783476613
	13	753550506	940607923	805656348
	14	760289320	951895825	762277027
	15	796339860	948585534	779122664
	16	721594203	941183306	749652430
	17	837183280	968910568	781626931
	18	762164356	951947470	747283372
	19	762644350	955107510	756603399
	20	753628856	959523532	773261726
	21	846490142	970984299	813460570
	22	744914917	946927834	755567148
	23	739540393	943009217	751304294
	24	765671910	938214869	746043406
	25	794517463	937234342	779733581
	26	728368846	953729215	749175289
	27	779444073	939791271	801641412
	28	741585288	947204715	798660086
	29	729618867	941746800	753179067
	30	725528234	956424369	755393926
Mean		754485152.5	951142676.2	770011621.5
Std. Dev		30935291.89	16171404.83	22173648.88
Confidence Interval		11551427.84	6038501.802	8279776.565

TABLE X
TIME TO PROCESS 101,821 WORDS USING SINGLE-THREADED PROGRAM ON VIRTUAL MACHINE SETUP

		Time to run (ns)
Trials	1	50324512
	2	45083897
	3	46009754
	4	44982723
	5	44518023
	6	44834643
	7	42945137
	8	44752789
	9	54026747
	10	43606507
	11	43791172
	12	44394384
	13	51029832
	14	43398933
	15	44553449
	16	44464350
	17	56781596
	18	45471425
	19	45153128
	20	45401803
	21	44718645
	22	44891162
	23	43222464
	24	45038465
	25	43253124
	26	45278998
	27	44904847
	28	47409402
	29	49214947
	30	43566076
Mean		45900764.47
Std. Dev		3245531.632
Confidence Interval		1211901.428

TABLE XI
TIME TO PROCESS 101,821 WORDS USING MULTI-THREADED PROGRAM (METHOD 1) ON VIRTUAL MACHINE SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	65418439	85635190	63330268
	2	59313678	76241551	61191076
	3	61466902	77273571	59616798
	4	77700221	79838117	60153410
	5	59629323	87544529	62208143
	6	67389730	76132413	59411851
	7	63704518	76176376	59575597
	8	64246569	75546087	62696051
	9	59078065	73533093	64923040
	10	56167495	78119881	61647581
	11	60086290	75013114	62786970
	12	57145159	74695668	63239175
	13	64301312	76200995	60464753
	14	68286276	79348404	57793633
	15	61473337	75350534	61414005
	16	55966470	81621504	63212293
	17	71039120	85641502	69644077
	18	67160953	79139702	64435417
	19	61072557	84224940	65258022
	20	57928132	74731028	62022713
	21	57988105	78091758	64918151
	22	62547868	78813511	64538946
	23	66424250	81049480	61630726
	24	61470505	76167528	64296607
	25	62078635	75253288	62235835
	26	60351936	87871345	66562978
	27	68356104	85351289	67904124
	28	60342034	75361773	82609846
	29	59861642	75653147	59606143
	30	57262638	79200652	66391542
Mean		62508608.77	78827399	63523992.37
Std. Dev		4866294.272	4189812.624	4504770.664
Confidence Interval		1817104.144	1564501.745	1682109.011

TABLE XII
TIME TO PROCESS 101,821 WORDS USING MULTI-THREADED PROGRAM (METHOD 2) ON VIRTUAL MACHINE SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	83435207	98197206	89411926
	2	76881952	90856473	76017358
	3	76657681	92298847	86583675
	4	80313198	81984376	91076044
	5	74748940	104451902	84179028
	6	70812339	92274345	77622936
	7	83162022	91144654	80796482
	8	75245025	86520610	107503095
	9	80728231	102185443	79167185
	10	73653379	90164483	85666796
	11	78361717	83769001	78897088
	12	77280760	86590930	78149115
	13	69170381	96206345	70163793
	14	80342428	90494075	88255348
	15	88769897	91877235	75001669
	16	78100417	98541599	73910404
	17	83211102	107371547	84952310
	18	81772223	88893171	75336995
	19	73187869	90210207	106195342
	20	75304339	89295238	79229012
	21	78864992	88599045	81057261
	22	77189351	90640311	79853291
	23	77316678	91117844	86153698
	24	82063374	95571605	80944127
	25	84978662	92544918	77998538
	26	79163611	98361363	76506295
	27	85137151	88818726	75255508
	28	76853452	90263703	74925630
	29	75177129	84246007	81775514
	30	77729633	95706678	74845300
Mean		78520438	92306596.23	81914358.77
Std. Dev		4367844.19	5875820.384	8453248.371
Confidence Interval		1630979.825	2194067.39	3156494.817

TABLE XIII
TIME TO PROCESS 1,660 WORDS USING SINGLE-THREADED PROGRAM ON VIRTUAL MACHINE SETUP

		Time to run (ns)
Trials	1	596906
	2	481631
	3	483825
	4	462646
	5	499326
	6	500312
	7	485527
	8	473292
	9	461660
	10	474410
Mean		491953.5
Std. Dev		39137.68237
Confidence Interval		14614.25077

TABLE XIV
TIME TO PROCESS 1,660 WORDS USING MULTI-THREADED PROGRAM (METHOD 1) ON VIRTUAL MACHINE SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	760926	771231	1291500
	2	809806	780138	1133482
	3	764943	881782	1229744
	4	798222	758120	1169747
	5	791576	806142	1288475
	6	759101	808909	1312581
	7	937041	807695	1323838
	8	927378	777413	1528864
	9	851495	819013	1140828
	10	988340	743539	1224483
Mean		838882.8	795398.2	1264354.2
Std. Dev		83406.06257	38999.1986	116080.7401
Confidence Interval		31144.3356	14562.54009	43345.26073

TABLE XV
TIME TO PROCESS 1,660 WORDS USING MULTI-THREADED PROGRAM (METHOD 2) ON VIRTUAL MACHINE SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	1377606	1071869	2165342
	2	1394190	1150727	2488307
	3	1415651	1131661	1784634
	4	1455631	993414	1582114
	5	1507684	1078139	2248019
	6	1592982	1023699	1719436
	7	1516599	1063622	2213255
	8	1553931	1107373	2432201
	9	1564555	1067377	1923545
	10	1806979	1072498	2031832
Mean		1518580.8	1076037.9	2058868.5
Std. Dev		125653.4822	46528.17511	303955.6891
Confidence Interval		46919.78135	17373.90612	113498.9196

TABLE XVI
TIME TO PROCESS 10,622 WORDS USING SINGLE-THREADED PROGRAM ON VIRTUAL MACHINE SETUP

		Time to run (ns)
Trials	1	3045407
	2	3092353
	3	2882037
	4	2843541
	5	2827001
	6	2837899
	7	2940013
	8	2837239
	9	3101034
	10	2822822
Mean		2922934.6
Std. Dev		114255.895
Confidence Interval		42663.85235

TABLE XVII
TIME TO PROCESS 10,622 WORDS USING MULTI-THREADED PROGRAM (METHOD 1) ON VIRTUAL MACHINE SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	4903197	5763439	5901496
	2	4936518	5545601	5724546
	3	4788872	4074098	10041281
	4	5877230	5180649	5354431
	5	5366764	5441415	5628620
	6	5579744	5392524	5186787
	7	5519045	5304545	5186965
	8	5085699	5470192	5248616
	9	5048370	5552999	5163177
	10	5160983	5181085	5649711
Mean		5226642.2	5290654.7	5908563
Std. Dev		347925.2112	462721.0252	1475673.089
Confidence Interval		129917.409	172782.8704	551025.3871

TABLE XVIII
TIME TO PROCESS 10,622 WORDS USING MULTI-THREADED PROGRAM (METHOD 2) ON VIRTUAL MACHINE SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	9180605	7495759	7744407
	2	7209162	7684450	10188016
	3	10537325	6573056	9961990
	4	9802288	9830503	9265662
	5	6776530	9200179	8072388
	6	6759667	7752337	6637679
	7	8089698	7702815	9721099
	8	8434628	7502438	6503218
	9	9275120	7764585	7815101
	10	7702978	7418983	10515914
Mean		8376800.1	7892510.5	8642547.4
Std. Dev		1300945.598	934276.6491	1475845.625
Confidence Interval		485781.07	348864.6342	551089.8135

TABLE XIX
TIME TO PROCESS 203,642 WORDS USING SINGLE-THREADED PROGRAM ON VIRTUAL MACHINE SETUP

		Time to run (ns)
Trials	1	87177099
	2	80580976
	3	81286834
	4	80589885
	5	82459330
	6	80992169
	7	85183747
	8	82534761
	9	83056409
	10	89071458
Mean		83293266.8
Std. Dev		2931781.085
Confidence Interval		1094745.049

TABLE XX
TIME TO PROCESS 203,642 WORDS USING MULTI-THREADED PROGRAM (METHOD 1) ON VIRTUAL MACHINE SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	114842425	167407970	107008215
	2	116902007	149467677	106262913
	3	111338200	141695930	106613606
	4	110321117	144589486	102474165
	5	107290839	146232545	101181310
	6	109441280	146443746	123011980
	7	110112664	138296980	112044838
	8	107267444	151866623	111105450
	9	103608746	143697331	112740054
	10	113963344	157024667	109232736
Mean		110508806.6	148672295.5	109167526.7
Std. Dev		3973234.542	8434579.858	6188459.012
Confidence Interval		1483630.161	3149523.88	2310808.572

TABLE XXI
TIME TO PROCESS 203,642 WORDS USING MULTI-THREADED PROGRAM (METHOD 2) ON VIRTUAL MACHINE SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	134649235	166123677	177961617
	2	142500088	167858115	140318334
	3	142588655	173965318	149165760
	4	139843085	184473498	136963007
	5	140023708	166510913	146475510
	6	139694987	167070909	138551203
	7	132401734	166950241	136503469
	8	141993927	171005624	134252595
	9	130346019	164958434	142774624
	10	133152246	177103539	142822620
Mean		137719368.4	170602026.8	144578873.9
Std. Dev		4612641.425	6219488.597	12603856.03
Confidence Interval		1722388.615	2322395.21	4706357.187

TABLE XXII
TIME TO PROCESS 509,105 WORDS USING SINGLE-THREADED PROGRAM ON VIRTUAL MACHINE SETUP

		Time to run (ns)
Trials	1	271063308
	2	219278408
	3	223887695
	4	212389347
	5	222374111
	6	219775424
	7	221478205
	8	223715299
	9	216105033
	10	240824468
Mean		227089129.8
Std. Dev		17156033.36
Confidence Interval		6406168.14

TABLE XXIII
TIME TO PROCESS 509,105 WORDS USING MULTI-THREADED PROGRAM (METHOD 1) ON VIRTUAL MACHINE SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	232512721	282380433	256576497
	2	235861929	278562088	277085607
	3	231794003	267036847	282974103
	4	240984074	292371722	291205572
	5	233533689	285256332	262625195
	6	245193473	267517803	279916752
	7	255264121	276683217	268656374
	8	235218114	264262700	281936179
	9	264507632	271780885	273458995
	10	275192296	268485721	278799871
Mean		245006205.2	275433774.8	275323514.5
Std. Dev		15109363.36	9204848.773	10281653.67
Confidence Interval		5641929	3437147.02	3839232.578

TABLE XXIV
TIME TO PROCESS 509,105 WORDS USING MULTI-THREADED PROGRAM (METHOD 2) ON VIRTUAL MACHINE SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	352289636	409009172	345676669
	2	345771011	393685355	348893770
	3	357860748	388869541	345227991
	4	336004276	397660811	338289528
	5	350023702	397947569	332619456
	6	356296810	399156999	365821778
	7	342010292	399389669	333603995
	8	344592309	386839773	349521245
	9	335096839	393525779	340761045
	10	341825263	400225954	334610054
Mean		346177088.6	396631062.2	343502553.1
Std. Dev		7862084.692	6301280.473	10007402.15
Confidence Interval		2935750.672	2352936.798	3736825.376

TABLE XXV
TIME TO PROCESS 712,747 WORDS USING SINGLE-THREADED PROGRAM ON VIRTUAL MACHINE SETUP

		Time to run (ns)
Trials	1	259176513
	2	260021229
	3	266369410
	4	252886711
	5	253120662
	6	258108565
	7	258136408
	8	260089949
	9	255504209
	10	253421388
Mean		257683504.4
Std. Dev		4164125.668
Confidence Interval		1554910.079

TABLE XXVI
TIME TO PROCESS 712,747 WORDS USING MULTI-THREADED PROGRAM (METHOD 1) ON VIRTUAL MACHINE SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	371784786	471941175	327021012
	2	358171785	458774242	342019051
	3	328759904	479753934	328396284
	4	337881630	475195791	331005260
	5	345879940	478665164	332688017
	6	356628283	469509278	341395949
	7	350473723	482575364	353458898
	8	346931588	483467424	374422512
	9	368419908	474621993	341946179
	10	379299907	468632641	347416103
Mean		354423145.4	474313700.6	341976926.5
Std. Dev		15677427.32	7482652.538	14256581.79
Confidence Interval		5854047.568	2794068.377	5323495.129

TABLE XXVII
TIME TO PROCESS 712,747 WORDS USING MULTI-THREADED PROGRAM (METHOD 2) ON VIRTUAL MACHINE SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	466915839	556788661	465484689
	2	450310939	553824924	469815841
	3	441855921	567328514	477928632
	4	445439515	552670083	467147227
	5	453812429	548148827	459222759
	6	434410102	542132553	450058041
	7	447762937	558201586	475928971
	8	452145340	546880780	472600575
	9	464683393	539700590	485388482
	10	431986961	521224904	484797094
Mean		448932337.6	548690142.2	470837231.1
Std. Dev		11397979.26	12593510.38	11034892.22
Confidence Interval		4256075.403	4702494.061	4120496.474

TABLE XXVIII
TIME TO PROCESS 1,019,210 WORDS USING SINGLE-THREADED PROGRAM ON VIRTUAL MACHINE SETUP

		Time to run (ns)
Trials	1	363510797
	2	353954865
	3	354931826
	4	372884979
	5	366151986
	6	360831270
	7	365608975
	8	376929855
	9	378300756
	10	360859811
Mean		365396512
Std. Dev		8443475.352
Confidence Interval		3152845.512

TABLE XXIX
TIME TO PROCESS 1,018,210 WORDS USING MULTI-THREADED PROGRAM (METHOD 1) ON VIRTUAL MACHINE SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	534102603	672556556	553615204
	2	502494954	662611777	541376824
	3	583898792	544507826	553745115
	4	522415767	536107807	545646604
	5	558571275	527675259	508346306
	6	512060834	580772252	485445523
	7	568802727	624191484	501754858
	8	495793758	561168028	493504427
	9	511469754	593325187	473892757
	10	532269794	499230748	540992515
Mean		532188025.8	580214692.4	519832013.3
Std. Dev		29534267.73	58046358.72	30394506.15
Confidence Interval		11028276.82	21674866.56	11349495.12

TABLE XXX
TIME TO PROCESS 1,018,210 WORDS USING MULTI-THREADED PROGRAM (METHOD 2) ON VIRTUAL MACHINE SETUP

		Time to run (ns)		
		4 Threads	2 Threads	8 Threads
Trials	1	684918016	878827480	706819027
	2	660500949	889380220	697299383
	3	644775215	968315315	697127740
	4	642748101	924296420	679506149
	5	693081579	904944043	726651966
	6	651647927	890355458	707911017
	7	638739851	922603779	705220374
	8	643129299	913623913	708609404
	9	671377636	872455503	709976901
	10	619561305	861974632	708939429
Mean		655047987.8	902677676.3	704806139
Std. Dev		22536038.93	31207699.59	12040616.98
Confidence Interval		8415095.234	11653146.54	4496040.27

TABLE XXXI
TEXTS USED FOR PROGRAM INPUT

Title	**Author**	**Word Count**
I Have a Dream... [6]	Martin Luther King Jr.	1660
A Modest Proposal [7]	Jonathan Swift	3397
The Cask of Amontillado [8]	Edgar Allan Poe	2354
A Tell-Tale Heart [9]	Edgar Allan Poe	2156
The Egg [10]	Andy Weir	1055
The Last Question [11]	Isaac Asimov	5463
I Have No Mouth And I Must Scream [12]	Harlan Ellison	5891
The Ones Who Walk Away From The Omelas [13]	Ursula K. Le Guin	2893
The Nine Billion Names of God [14]	Arthur C. Clarke	2619
The Star [15]	Arthur C. Clarke	2477
The Lottery [16]	Shirley Jackson	3450
On Seeing The 100% Perfect Girl One Beautiful April Morning [17]	Haruki Murakami	1475
Disappointing: Microsoft Confirmed That Project Scarlett Is Actually Just A Brothel They're Building In Thailand Where Xboxes Can Go To Be Pleasured [18]	The Onion	403
'Pokémon' Fans Are Up In Arms With Game Freak's Refusal To Include A National Sex Offender Registry In 'Sword and Shield [19]	The Onion	384
Making Amends: Blizzard Added A Drawing Of Xi Jinping Getting Pinched On The Ass By A Crab To All Spawn Rooms On Overwatch's Lijiang Tower Map [20]	The Onion	360
Ich Bin Ein Berliner Speech [21]	John F. Kennedy	676
A Date Which Will Live in Infamy [22]	Franklin D. Roosevelt	503
Nightfall [23]	Isaac Asimov	13218
How To Lose Weight in 4 Easy Steps [24]	Aaron Bleyaert	1858
Cousin Tribulation's Story [25]	Louisa May Alcott	819
The Story of an Hour [26]	Kate Chopin	1003
Ali Baba and the Forty Thieves [27]	Arabian Nights	8820
The Call of Cthulhu [28]	H.P. Lovecraft	11761
The Monkey's Paw [29]	W.W. Jacobs	3931
The Brave Tin Soldier [30]	Hans Christian Andersen	1736
The Haunted Mind [31]	Nathaniel Hawthorne	1768
Araby [32]	James Joyce	2328
Regret [33]	Kate Chopin	1438
Gift of the Magi [34]	O. Henry	2069
Ex Oblivione [35]	H.P. Lovecraft	702
The Game [36]	Mary Roberts Rinehart	10316
On the Day of the Crucifixion [37]	Leonid Andreyev	1452
A Dead Woman's Secret [38]	Guy De Maupassant	1395
Total Word Count	**101821**	