PA1 – Ping Pong Documentation

Brian Conway

23 Feb. 2017

CS415

Methodology

The file pingPong.cpp sends a single integer back and forth between two processes 500 times, timing from the time the integer is sent to the time the integer is received back from the other process. It divides the total time taken for all messages by 500 to get the time taken for a single ping pong. It then stores the time and repeats this process 1000 times. The number of messages back and forth and the overall iterations can be adjusted by changing the two constants at the beginning of pingPong.cpp.

After the 1000 readings are taken, the calcStatistics function finds both the average and the standard deviation of the ping pong times. It outputs this information to the console and then outputs the statistics and the measurements to a file called “measurements.csv”. The first line of the output file is the average, the second is the standard deviation, and the third are all the measurements separated by commas.

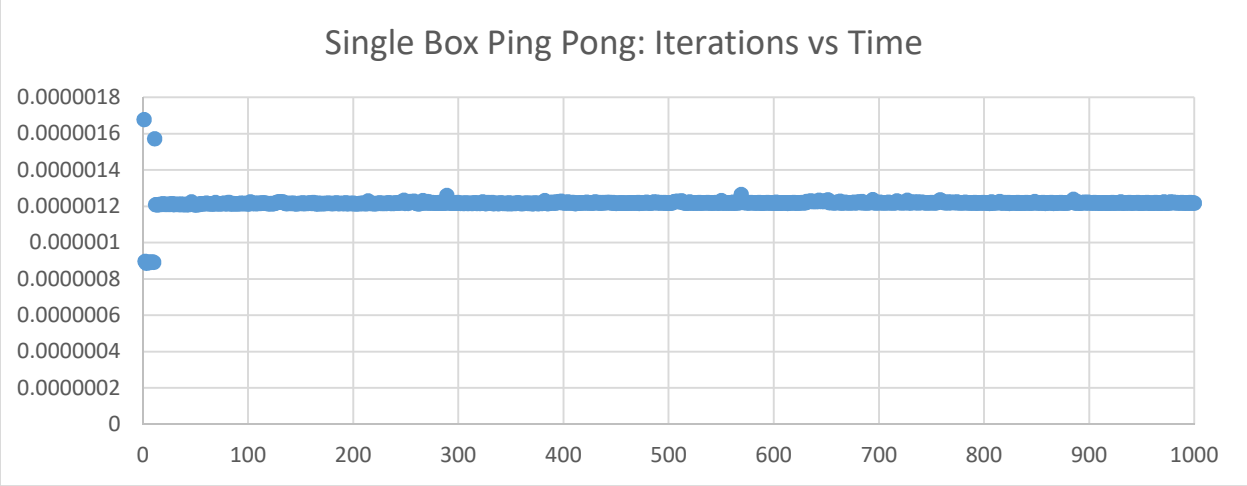
The file bufferChecker.cpp sends an array of integers back and forth between two processes 10 times, timing from the time the array is sent to the time the array is received back from the other process. It divides the total time taken for all messages by 10 to get the time taken for a single ping pong. It then stores the time and repeats this process 10000 times, with the array starting at size 1 and incrementing its size by 1 each iteration. The number of messages back and forth and the overall iterations can be adjusted by changing the two constants at the beginning of bufferChecker.cpp. Measurements are output in “measurements.csv”.

Timing is done using the custom Timer class, which has start, stop, and resume functions similar to a stopwatch. Timer makes use of timevals and the gettimeofday() function in order to get an accurate reading of seconds and milliseconds at certain instants. When the elapsed time is to be given, the Timer takes the difference in seconds and milliseconds between when the timer was started and when the timer was stopped. It then combines the seconds and milliseconds reading into a double precision floating point variable.

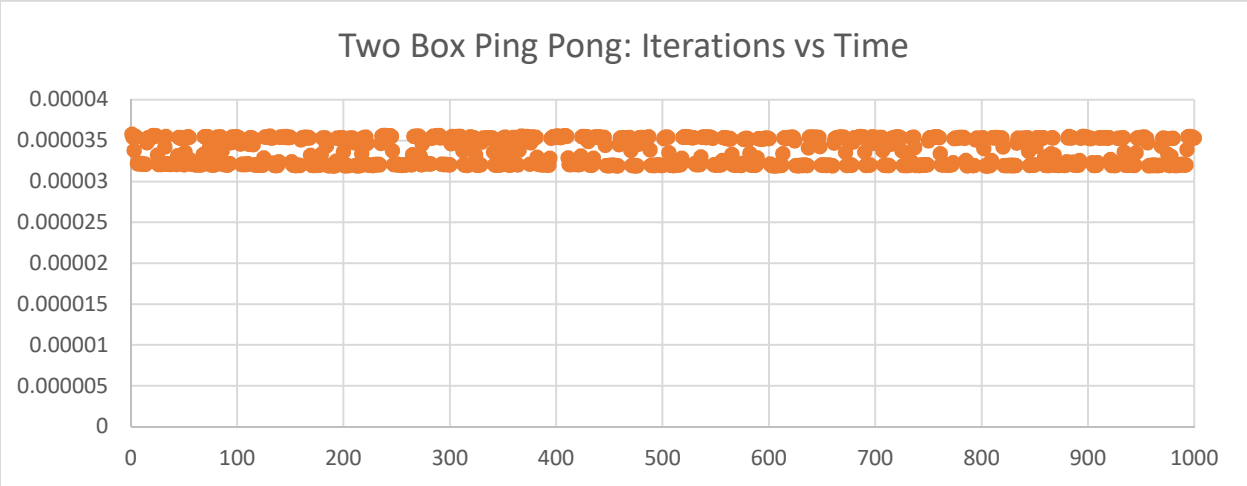
Results

In order to increase accuracy, the messages sent in each ping pong was increased to 1000 for both the ping pong and the buffer checking. The values were decreased in the submitted code in order to cut down on runtime.

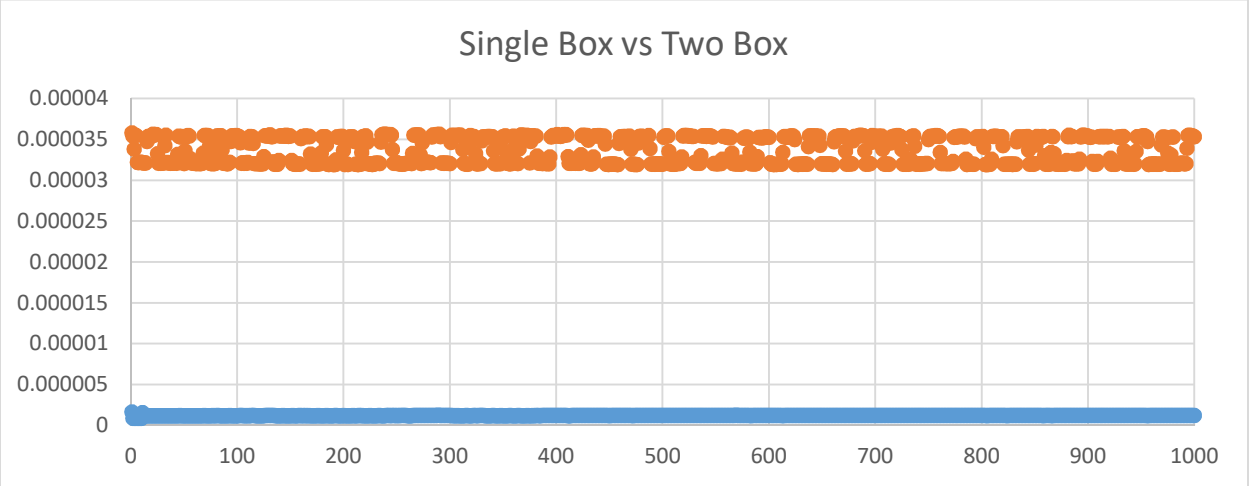
Below is a graph of the ping pong times for two processes on the same machine. The average of 1000 iterations was 0.00000122 seconds (1.22 microseconds) with a standard deviation of 0.0000000362 seconds (0.0362 microseconds).



Below is a graph of the ping pong times for two processes on two machines. The average of 1000 iterations was 0.0000337 seconds (33.7 microseconds) with a standard deviation of 0.00000179 seconds (1.79 microseconds).



This graph is a comparison between the two, with the single box in blue and the two box measurements in orange.



The next graph shows the measurements for the timings of the arrays being sent back and forth. The x-axis represents the number of integers sent between the processes and the y-axis represents the time taken to ping pong that array of integers. At around 1000 integers, the time taken for the ping pong doubles in size. This leads me to believe that is the point where the number of integers exceeded the buffer size.