Multithreading and Hash Function Performance

Andre M. Tudor and Ege B. Konya

Department of Computer Science, The University of Texas at Dallas

CS3377.505: Systems Programming in UNIX and Other Environments

Professor Sridhar Alagar

April 14, 2023

Multithreading and Hash Function Performance

The following experiment tests the relationship between the performance of the Jenkins one_at_a_time hash function relative to the number of threads it is performed on. Each specified input file– each of increasing size– is divided among the threads to be hashed in parallel. The cumulative results for each file will be graphed and compared to analyze for performance increases with regard to the number of threads.

Experiment

Experimental Environment

The experiment data was gathered remotely on cs3.utdallas.edu. The server is an x86_64-bit machine with 48 Intel(R) Xeon(R) CPU E5-2695 v2 2.40GHz CPUs.

Methodology

The test program was executed on each test file a total of seven times with an increasing number of threads each iteration (1, 4, 16, 32, 64, 128, 256). The resulting time of each execution was recorded. For every test file, two ways to visualize the results were given. First, the resulting times were graphed with regard to the number of threads it was run on. Second, the factor of speedup (time taken for a single thread divided by the time taken by n threads) was graphed with regard to the number of threads.

Figure 1

Time (s) vs. Number of Threads for p2_tc0

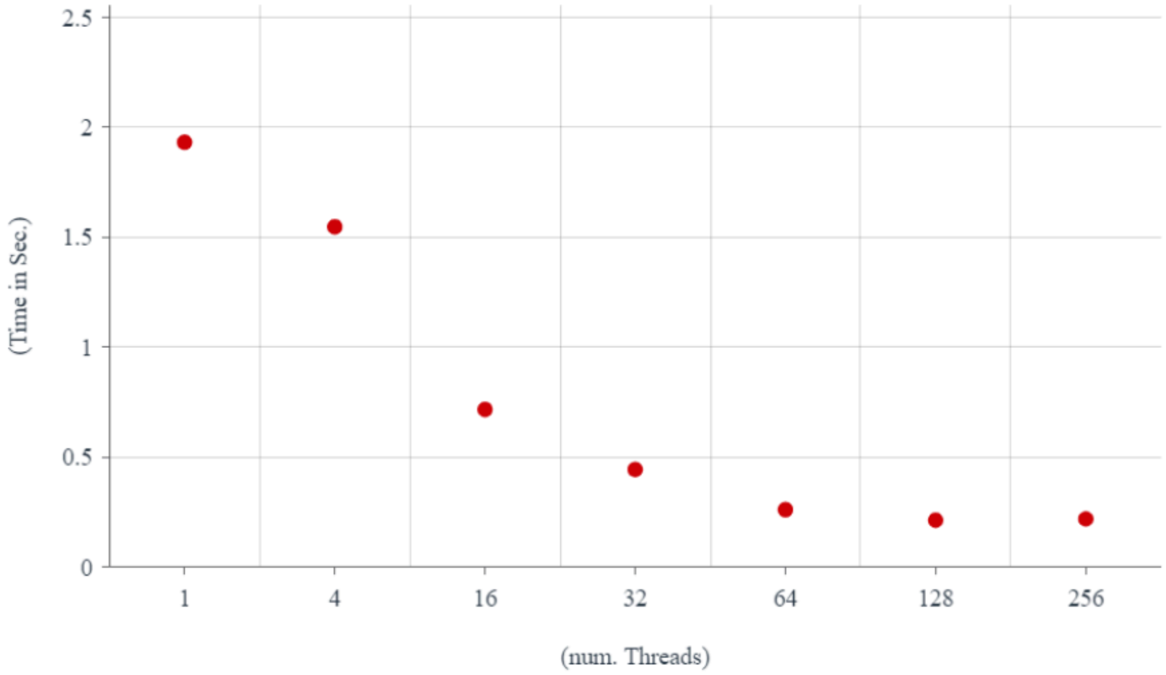


Figure 2

Speedup Factor vs. Number of Threads for p2_tc0

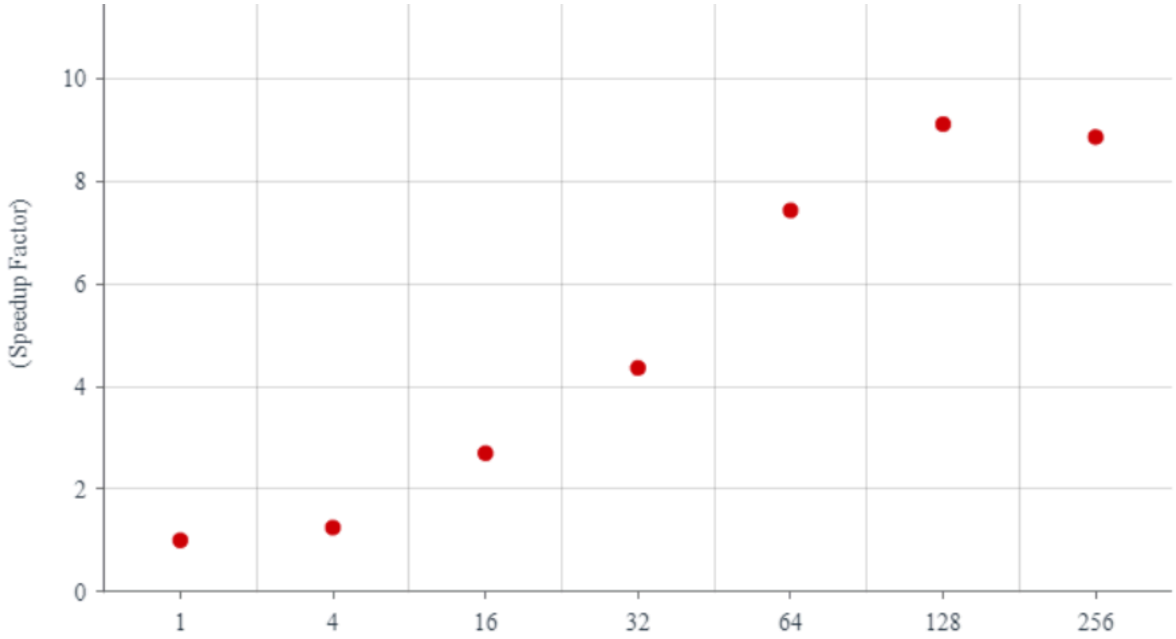


Figure 3

Time (s) vs. Number of Threads for p2_tc1

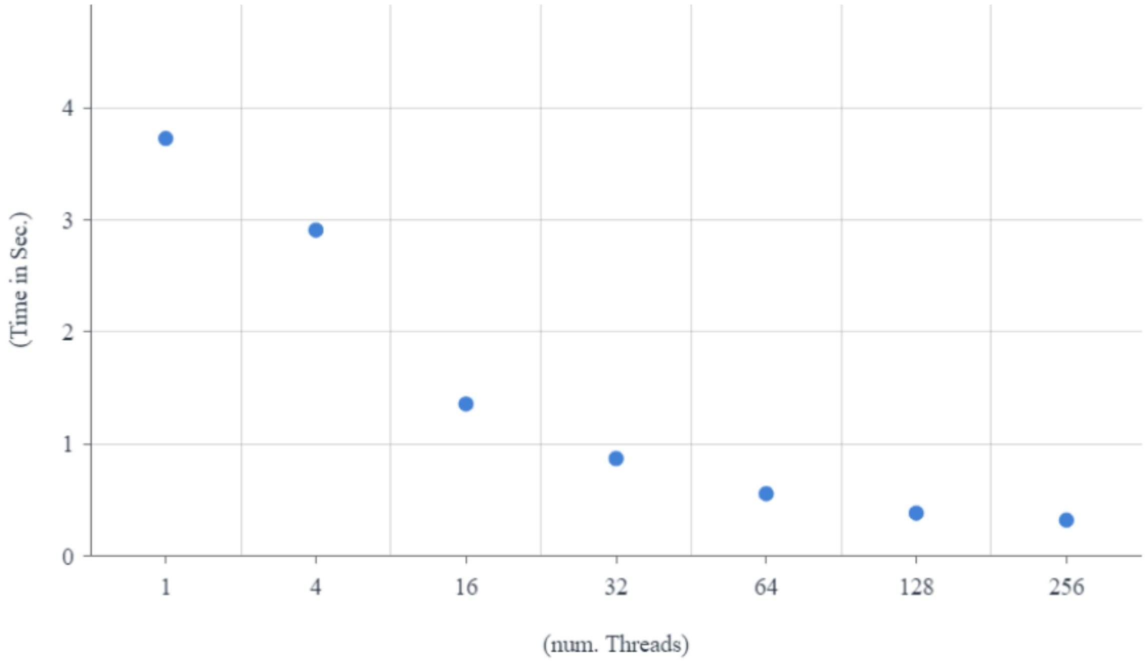


Figure 4

Speedup Factor vs. Number of Threads for p2_tc1

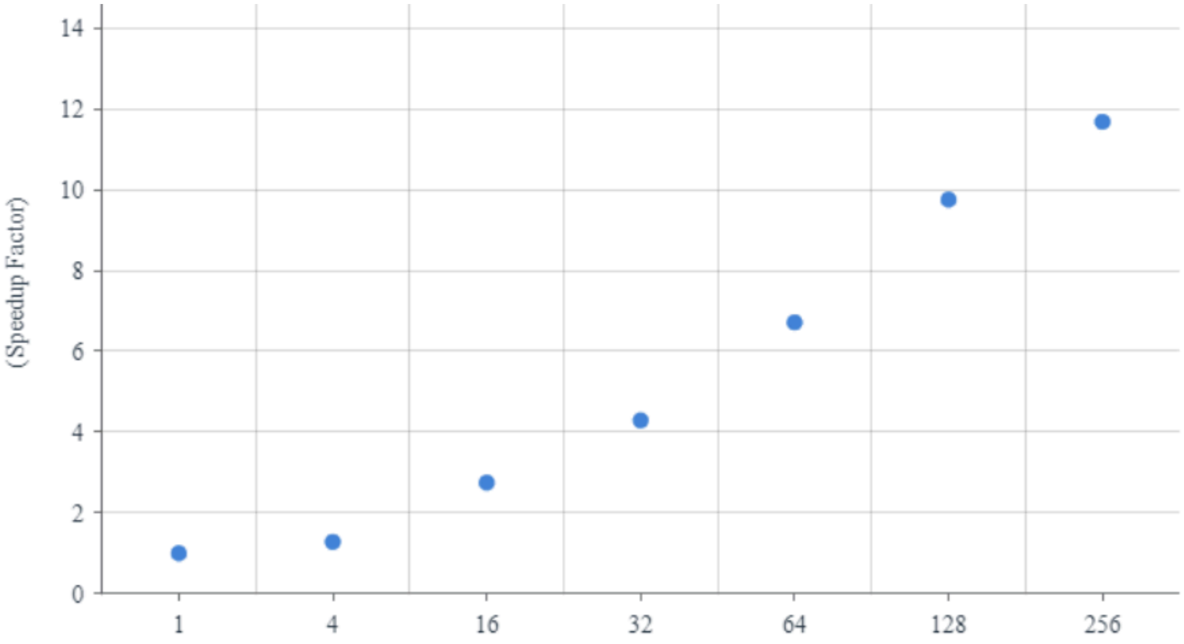


Figure 5

Time (s) vs. Number of Threads for p2_tc2

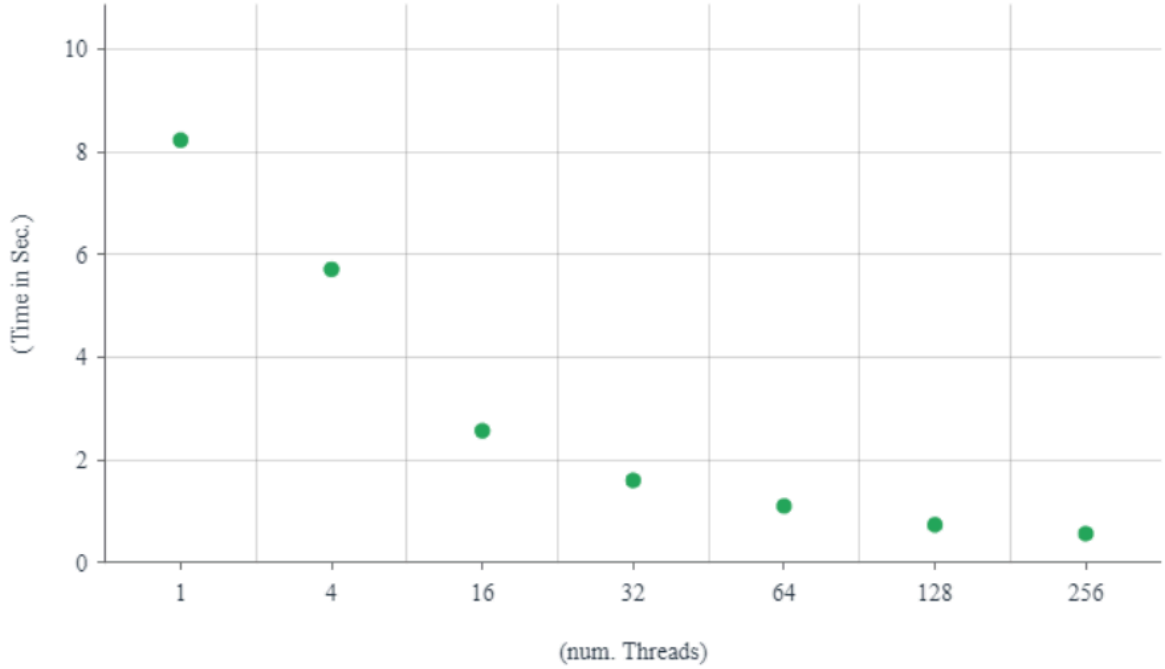


Figure 6

Speedup Factor vs. Number of Threads for p2_tc2

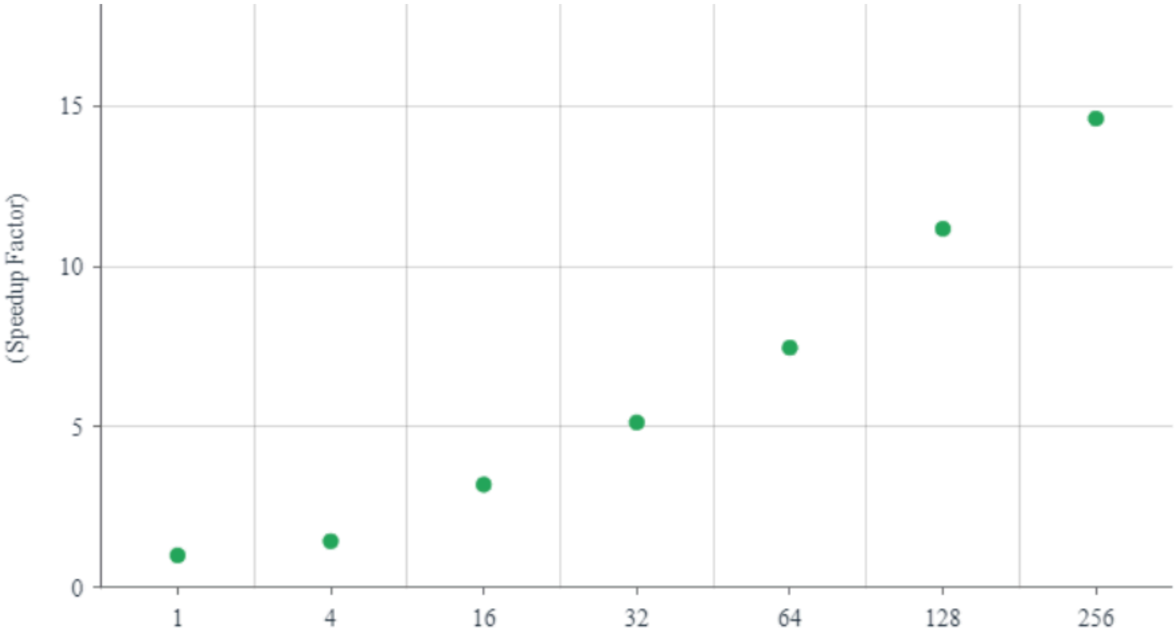


Figure 7

Time (s) vs. Number of Threads for p2_tc3

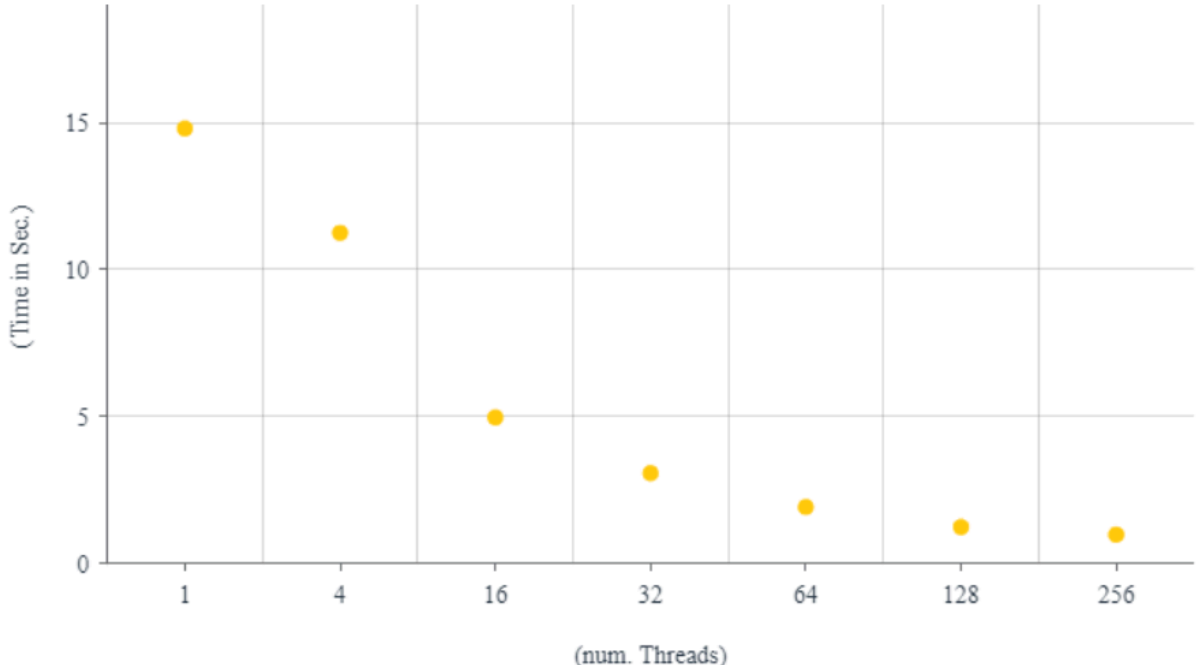


Figure 8

Speedup Factor vs. Number of Threads for p2_tc3

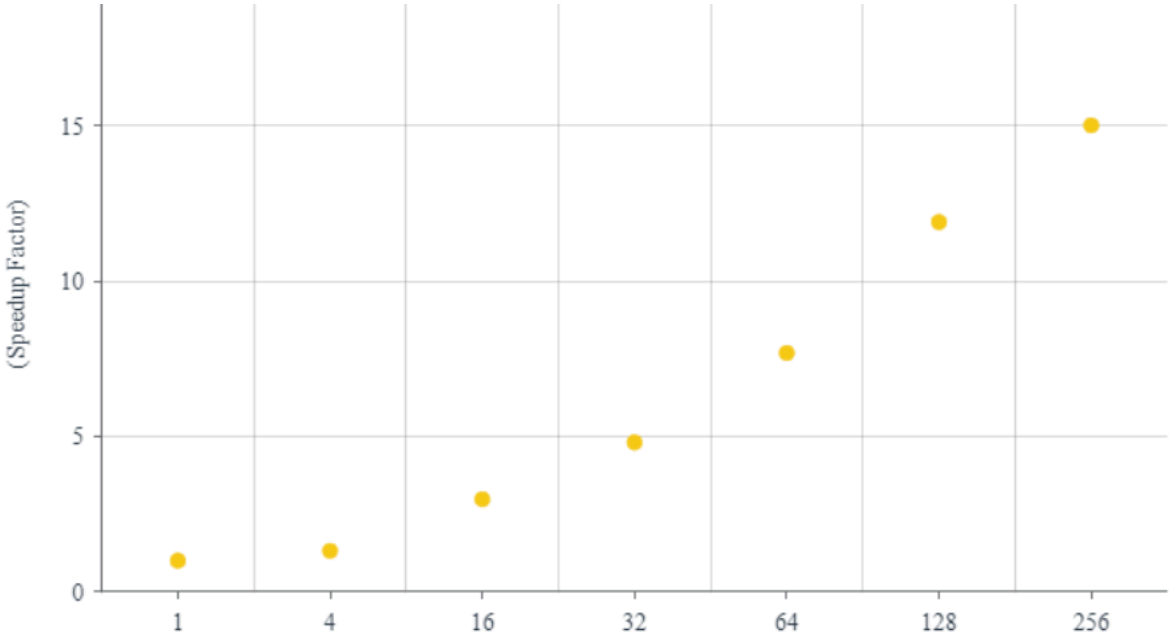


Figure 9

Time (s) vs. Number of Threads for p2_tc4

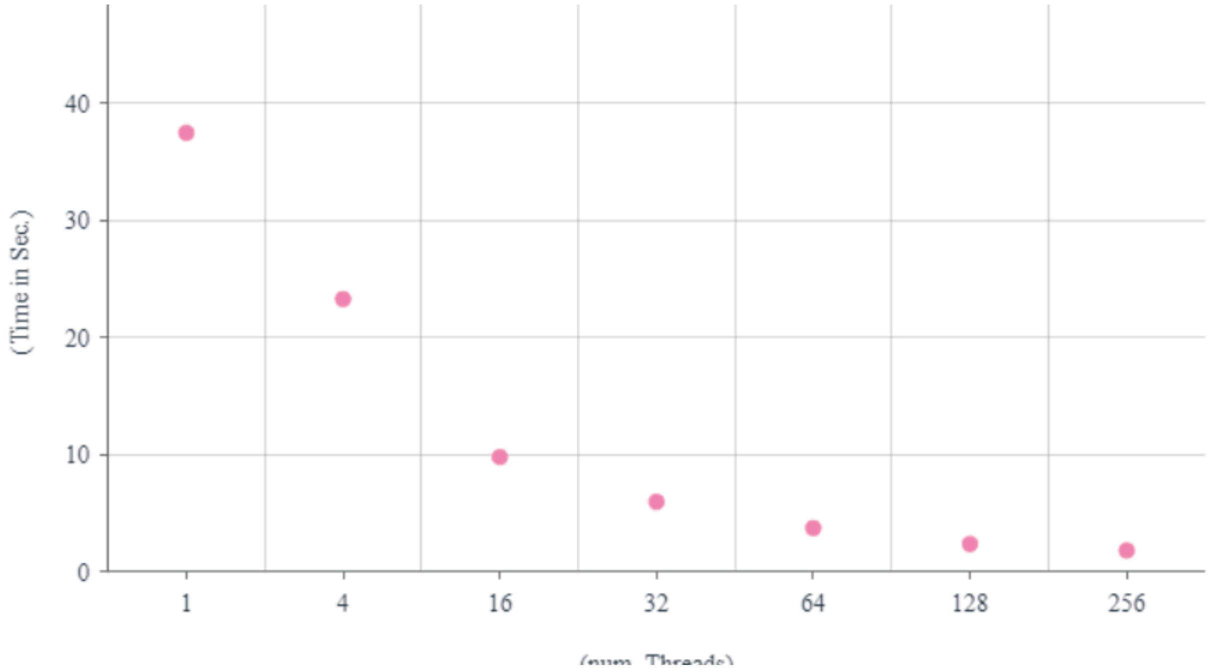
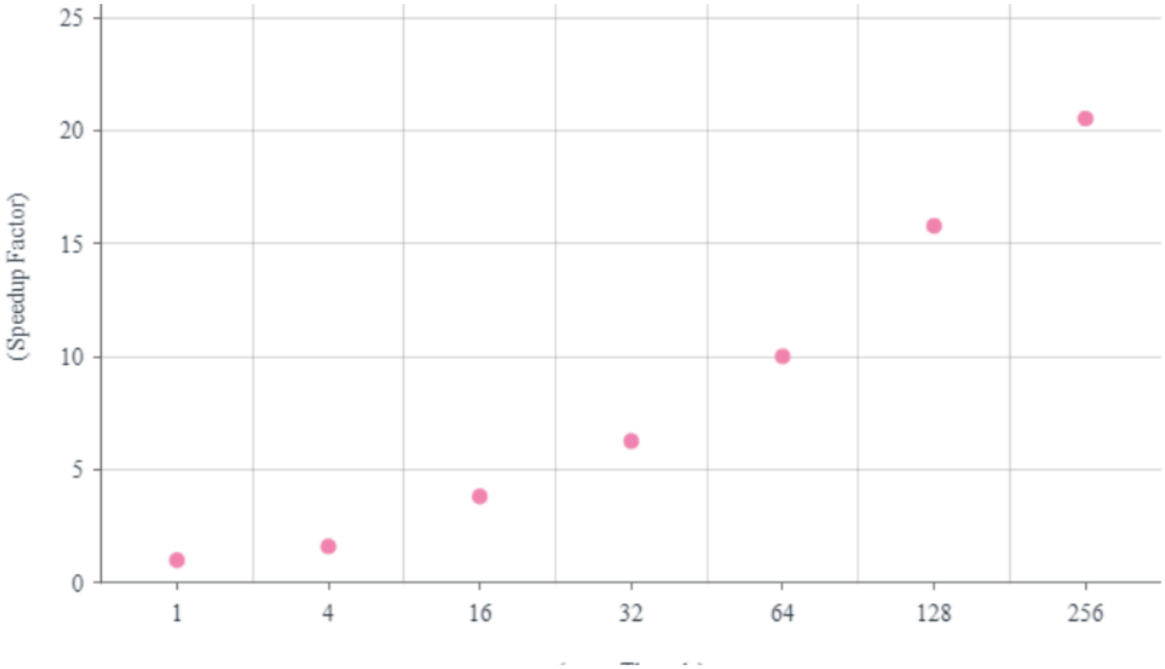


Figure 10

Speedup Factor vs. Number of Threads for p2_tc4



Analysis

After testing using different file sizes and a different number of threads, the results clearly indicate that when the number of threads increases the expected time taken to compute the hash value decreases. This relation, however, cannot be extrapolated to any number of threads. Every file size will reach a point where any benefit incurred by more threads is immediately negated by the necessary overhead of creating, executing, and joining such threads. The factor of speed-up achieved when increasing the number of threads represents a relatively linear relation to the amounts of threads our experiment studied. The factor itself is determined by the size of the file being tested on (i.e. greater file sizes result in larger factors of speed-up). This would indicate that the benefits of a multithreaded hashing process significantly increase when used on files of a larger size.