CALIFORNIA STATE SCIENCE FAIR
2012 PROJECT SUMMARY

Name(s)
Andrew C. Haden

Project Number
S1407

Project Title
The Effect of Parallel Processing on MD5 Brute Force Efficiency

Abstract

Objectives/Goals
My project aimed to find what level of parallel processing, quantified in threads, results in the best performance of an MD5 brute force algorithm determined by permutations tested per second and CPU load.

Methods/Materials
In my project, I created a program in the C# language that executes a brute force algorithm with provided numbers of threads. I ran this program on a Windows-based computer with a four-core processor.

Results
The permutations tested per second and CPU load (%) increased from 54,958 and 14% at one thread to at 141,687 and 53.6% at four threads, the peak. A significant performance decline was observed with only 34,935 permutations/second and 17.6% CPU load at eight threads and 25,588 permutations/second and 15.4% CPU load at 16 threads.

Conclusions/Discussion
The performance of the MD5 brute force peaked at four threads with 141,687 permutations tested per second and 53.6% CPU load. The performance was negatively affected when the algorithm was run with 8 and 16 threads. From this, I concluded that the performance of an MD5 brute force algorithm is greatest when parallelized with the same number of threads as the host computer has CPU cores. I assume this also applies to other computationally-heavy tasks that are parallelized, however that would have to confirmed in a separate study.

Summary Statement
My project found what level of parallel processing, in threads, results in the best performance of an MD5 brute force algorithm.

Help Received
None