# An Analysis of Jacobi Iterations vs Thread Count

By: Nat Choeypn and Jason Lacy

**Introduction:**

Efficiency is quickly becoming one of the most desirable traits in technological programs due to the ever-increasing demand for bigger and bigger data. Over the last two years alone, we have generated about 90% of the total data that has ever existed on the planet earth. A mind-boggling 2.5 quintillion bytes of data are created each day, and this number is only getting larger (Forbes.com).

In this age of Big Data, users do not care how we retrieve their Facebook status updates; they care only care that its reliable, secure, and accessible. They don’t care how many softball games the programmer missed trying to finish his SQL queries if Snapchat decides to crash mid-selfie because an overworked thread couldn’t handle the pressure. We, as programmers, are here to solve problems. We must go beyond the call of duty when creating a program that technologically illiterate users will utilize daily. With the ever-decreasing attention spans of today’s youth, algorithm speed and data efficiency are almost as important as getting the correct output. These are the reasons why we need to learn how to write programs that function quickly and correctly. Cost-effective multithreading gives us the tools we need to increase the speeds at which we compute. We need to think past what is thought of and continue to innovate in every area imaginable.

**Implementation:**

Blah

**Experiment:**

Blah

**Results:**

Blah

**Conclusion:**

Blah