# Project 3: Multithreaded MapReduce

Almost everyone has heard of Google’s MapReduce framework, but very few have ever hacked around with the idea of map and reduce. These two idioms are borrowed from functional programming, and form the basis of Google’s framework. Read: <http://developer.yahoo.com/hadoop/tutorial/module4.html> to understand what MapReduce and Hadoop are. You may also read the following article (on Canvas): *MapReduce: Simplified Data Processing on Large Clusters*, by Jeffrey Dean and Sanjay Ghemawat.

The Python[**multiprocessing**](https://docs.python.org/3/library/multiprocessing.html) package supports spawning processes using an API similar to the threading module. The multiprocessing package offers both local and remote concurrency using subprocesses instead of threads. Due to this, the multiprocessing module allows the programmer to fully leverage multiple processors on a given machine.

The Python multiprocessing.Pool package provides a mechanism for the parallelization of map/reduce style calculations.

Read:

<https://www.youtube.com/watch?v=_1ZwkCY9wxk>

<https://chryswoods.com/parallel_python/mapreduce_part2.html>

**Problem:** Find the frequencies of title-cased words (proper nouns, starts of sentences) in a very large text file. Use Python multiprocessing pool.

**Specifications:**

* Use benchmarking to compare the efficiency of the program for different values for *n* (number of words of the input file).
* Try to scale up the number of threads.