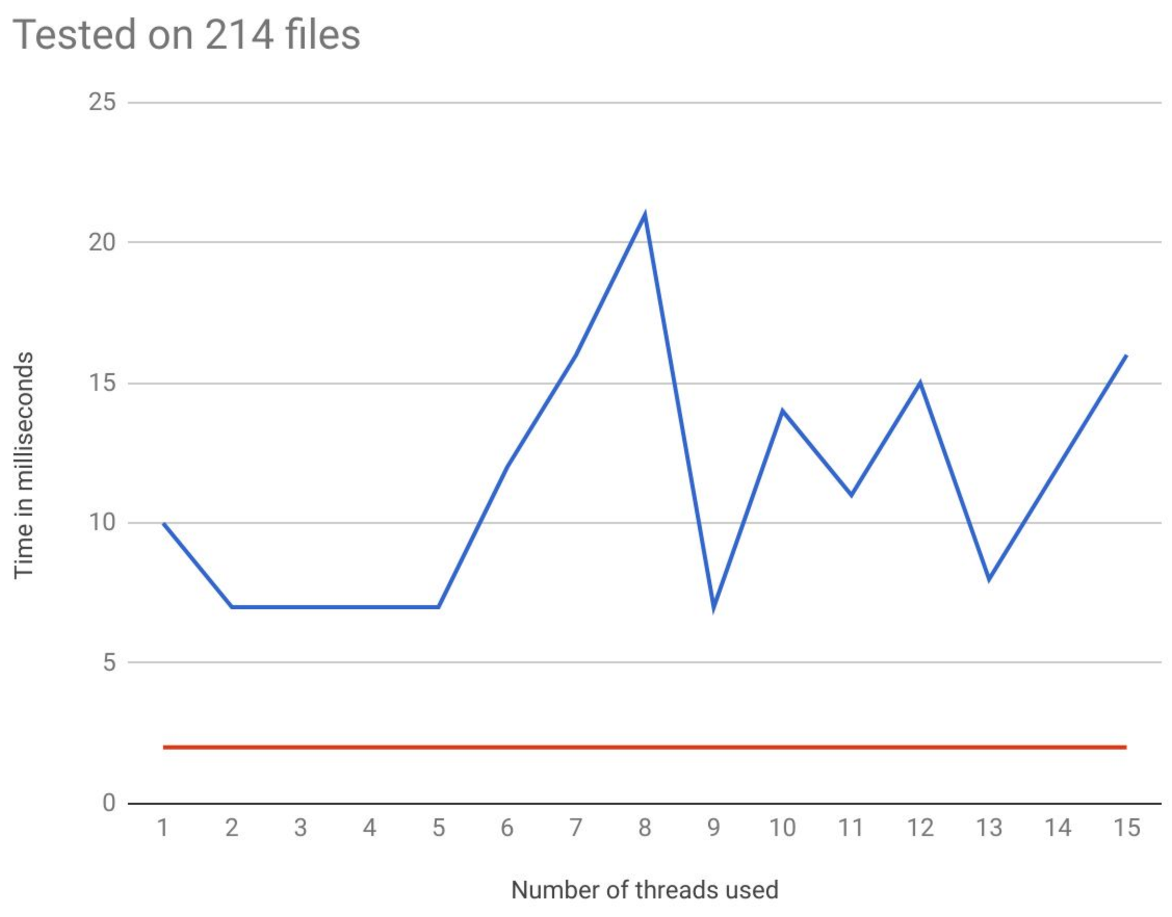
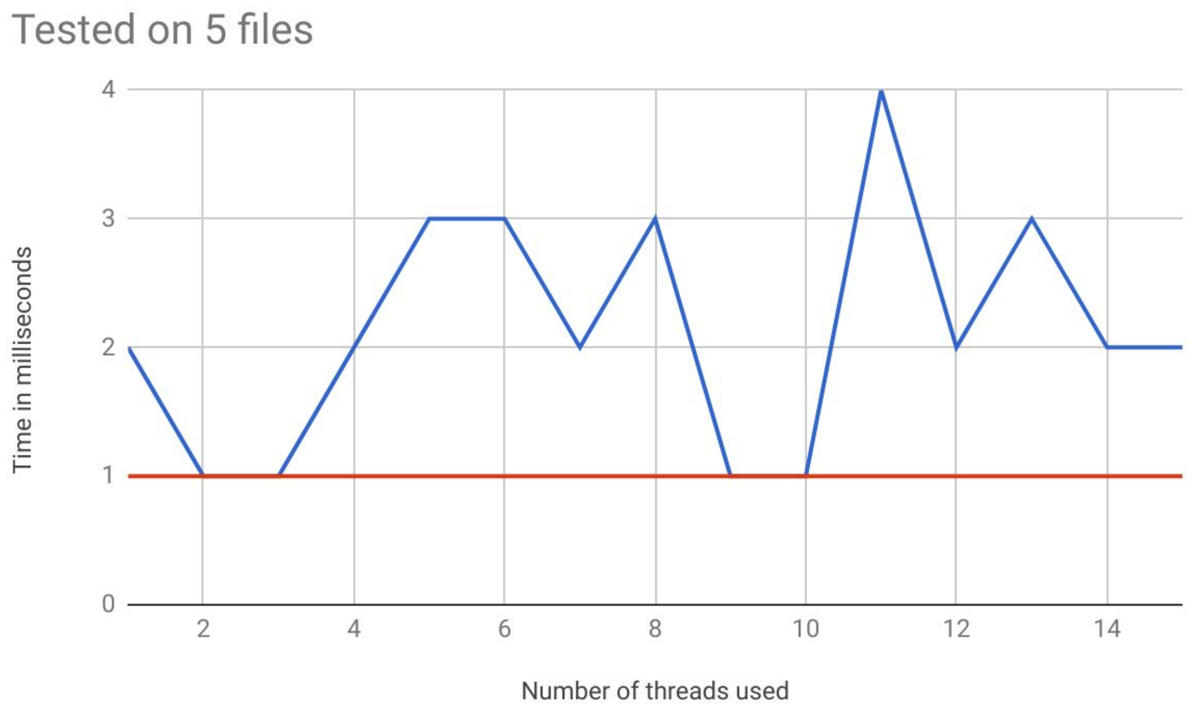
Time Analysis

The purpose of that project was to input a list of file names and gather some statistics for all the files.The last implementation we had to do was to measure the time for both the serial and the multi threaded architecture.From a theoretical perspective it would be more efficient to to split the file names to up to 15 threads rather than running them one after the other in the main thread.Actually we could run code at the same from different threads only if we had cores up to the numbers of the threads we use.In this project we run the code on a dual core machine.



The practical results are contradictory to our theoretical speculation.
This can be explained due to the fact that the procedure that has to be executed by each thread is very trivial and probably the thread creation and the thread joining take more time than the procedure.



This second testing gives ground to our second speculation (that the procedure that has to be executed by each thread is very trivial and probably the thread creation and the thread joining take more time than the procedure) , showing that for a lot fewer files the multi threaded approach will be a lot closer to to the the serial design, since not as many threads have to be created and joined.