This document acts as a summary of timing studies using the Clojure programming language to demonstrate the effects of concurrent execution as a way of enhancing performance.

Problem statement:

Write a program in Clojure that will read a text file containing unsigned integers (which will be provided later). Your program will read in a large collection of integers and put them into a list. Using code in your program (i.e. NOT calling a library routine), sort the integers into order, using either the quicksort or mergesort algorithm. For the first pass, carry this out in a single-threaded program. Then, using Clojure's parallelism options, repeat the sort of the original list, using 2, 4, 8, 16, and 32 threads. Repeat all sorts 5 times on the same hardware and report the average times. Do not count file access time as part of the sorting time. Plot the completion time as a function of the number of threads...

Running on Linux Mint 19 “Tara” x32 bit, Intel Core M-5Y10c CPU @ 0.80GHz Quad-Core, 2048MB RAM.