Assignment 3

Concurrency in Clojure

Part A. [30]

1. Consume from a sequence of *N* random strings.
2. For each string, count the number of distinct characters.
3. Aggregate the total count for the entire sequence.
4. Record the time it takes to complete the task.
5. Plot the relationship between *N* and the total runtime *t*.

Your source code *must be named* **“sequential-count.clj”**.

Part B. [60]

In this part, you will break the task into *K* threads using the **(future …)** form. The parallelizing strategy is to partition the sequence of strings into *K* subsequences. Each subsequence will be counted separately using (future …). Then, add up the individual counts by deferencing each of the *futures*.

1. Record the total time it takes.
2. Plot the relationship between *K* and the total runtime *t*, with *N > 1,000,000*.

Your source code *must be named*  **“future-count.clj”.**

Report [10]

1. Submit your source code files with the correct names.
2. Submit your report, named **report.pdf**. It must contain the two plots.
3. For each observation, discuss the effect of concurrency and the choice of *K*.

Part C. [10 Bonus]

Devise your own parallelizing strategy to utilize *software transactional memory* (STM). While using STM features, explore the possibility of optimizing your concurrent updates of a shared counter using atoms.

Your source code must be named **“stm-count.clj”.**

Part D. [10 Bonus]

Solve the same problem using *channels and go-expressions*.

Your source code must be named **“go-count.clj”**.

Part E. [10 Bonus]

Solve the same problem using *agents*.

Your source code must be named **“agent-count.clj”**.