CS6868 - Concurrent Programming

Programming Assignment - 3

Due: 20th March 2013

This exercise builds upon the previous lab assignment. The exercises below should be implemented using Cilk. As before, ensure that the sequential and the parallel versions of the program produce the same output. Use `-cilk-serialize` and ensure that the results are right.

The final deliverable is a single program that chains together the work done by the four modules written previously.

Specifications

Command Line Arguments

Argument Number	Value
1	Number of rows (**r**)
2	Number of columns (**c**)
3	Value range for matrix (**k**)
4	Shuffle input file (**shuffle_in**)
5	Percentage of pixels for thresholding (**p**)
6	Number of steps for Game of life simulation (**n_steps**)
7	Common prefix of output file names (**out_template**)

Constraints

- r, c and k are at most 10000.
- Input file could be read ahead of time so that the rest of the program has only pure transformations.
- *n_steps* is at most 1000.

Operations to be carried out

- Create a random matrix M of size $r \times c$ filled with values in range of $1 \ldots$ k
- Shuffle the matrix M based on the operations specified in the `shuffle_in`
- Threshold the matrix M of integers and generate the binary image B such that B_{ij} is 1 if no more that $p\%$ of pixels in M are greater than M_{ij}
- On the thresholded matrix B, run `n_steps` steps of Conway's game of life simulation. Refer the second assignment for rules pertaining to the game of life
- Optionally, have an option (`#define`d) to dump the intermediate steps in the game of life into files with named `out_template_StepNumber`
- When reporting numbers, do not dump these intermediate files. Dump only the result at the end of the last step with name `opt_template_LastStepNumber`

  Note that the `opt_template` and `StepNumber,LastStepNumber` are all variables. The template is taken as a command line argument and the other two are computed.

Deliverables

- Sources for the program
- A README specifying the build procedure and special thing to look out for when compiling/running the program
- A report that contains the following information
  - Difference in runtime of the new program and the runtime achieved by chaining together the programs of assignment 2.
  - Similar results for peak memory consumption
  - The approach taken by you for solving the problem highlighting any parallelization achieved between successive stages of the process and other strategies used.
  - Any other observations made which you would find interesting
- The report should include results for 1000×1000 and 10000×10000 matrices with 1000 shuffles, and $p = 10\%$