Prof. Jingke Li (FAB 120-06, li@cs.pdx.edu); Class: MW 14:00-15:50 @ UTS 308; Office Hr: M 13:00-14:00 & by appt.

Assignment 4: Programming with Chapel
(Due Wednesday, 6/1/16)

This assignment is to practice programming in the new parallel programming language, Chapel. You'll write data-parallel shared-memory programs first, and then use the domain map feature to convert them to PGAS-style distributed-memory programs.

For this assignment, all students will write the same set of programs. However, CS515 students are required to implement one extra set of functions within these programs (see below). This assignment carries a total of 10 points.

If you haven't done so, run `addpkg` on the Linux system to add Chapel 1.12.0 to your environment, and set the environment variables required by the language. (See Lab1 handout for details if you need.)

Download and unzip the file `assign4.zip`, you'll see a directory `assign4` containing the following program files:

- `jacobi.c` — a sequential Jacobi program in C
- `jacobi-shm0.chpl` — a starter version of a shared-memory Jacobi program in Chapel
- `Makefile` — for compiling programs

Your Tasks

This assignment consists of four tasks.

CS415 Students: Anywhere "two versions of Gauss-Seidel iteration" is mentioned, you only need to implement one of them, of your choice. You may also choose to implement both, for which you will earn up to 20% extra points.

1. The file `jacobi.c` contains a function, `jacobi()`, which implements the Jacobi iteration for Laplace's Equation. Your task is to write two functions in the same style to implement the Gauss-Seidel iteration:
   - `gauss_seidel()` — Gauss-Seidel iteration with natural index ordering
   - `red_black()` — Gauss-Seidel iteration with red-black ordering

   Copy `jacobi.c` to a new file, `laplace.c`; add the two new functions to it; and modify the `main()` function to include additional calls to the new functions. Run and test your program. Obverse the convergence rates of the different iteration methods.
2. The file `jacobi-shm0.chpl` is a starter version of a shared-memory program for the Jacobi iteration, written in Chapel. Read and understand the given program part first. Your task is to complete the function `jacobi()` with data parallel and/or shared-memory constructs. Note that the computation in the Jacobi iteration are defined only for the interior points of the domain. Rename your program to `jacobi-shm.chpl`. Run and test it.
3. Write two functions using the same signature as the function `jacobi()` to implement the Gauss-Seidel iteration. Copy `jacobi-shm.chpl` to a new file, `laplace-shm.chpl`; add the two new functions to it; and modify the `main()` function to include calls to the new functions. Run and test your program.
4. This last part is to convert your shared-memory program, `laplace-shm.chpl`, to a PGAS-style distributed-memory program, `laplace-distr.chpl`, by using Chapel's domain map feature. Both the Jacobi function and the Gauss-Seidel function(s) need to be converted.

Use block distribution for the domain map. Think clearly which domain(s) should be used as the base for domain map. Note that the program structure and the computation steps do not need to change.

You need to find a way to verify that the mesh point updates are indeed being computed distributively according to the block distribution. Include proper `writeln` statements in the program for this purpose. (You may want to guard them with a `verbose` flag, which you can control to turn on and off with a runtime switch.)

What to Turn In

As before, write a two-page summary documenting your experience with this assignment. Did you observe different convergence rates between Jacobi and Gauss-Seidel methods? Are the results consistent across the sequential, the shared-memory, and the distributed-memory versions? How do their performance compare? etc.

Make a `zip` file containing your three programs, `laplace.c`, `laplace-shm.chpl`, `laplace-distr.chpl`, and the summary report. Use the Dropbox on the D2L site to submit your assignment file.