Prof. Jingke Li (lij@pdx.edu); Classes/Labs: TR 1400-1550 @FAB 150, 88-10; Office Hours: TR 1300-1400 @FAB 120-06

Lab 9: Chapel Programming (Part 2)

1 Domains, Arrays, Locales, and Domain Maps

1. Read and understand the demo programs, `domain1.chpl` and `domain2.chpl`. Pay attention to the connection between array `a` and domain `D`. Compile and run the programs; notice how the changes on domain `D` affect array `a`. If you want, you may change the programs and see the effects.
2. Read and understand the demo program `locales.chpl`. Compile and run it. Observe the output.
3. Read and understand the demo program `domainmap.chpl`. Compile and run it. Change some mapping parameters in the program and observe the effects.

2 Matrix Multiplication

The file `mmul.chpl` contains a matrix multiplication program in Chapel. Convert the program to a PGAS version, `mmult2.chpl`, by introducing domain maps. Add the following code to verify that array `c` is indeed partitioned across the locales:

```
write("Locale:");
for (i,j) in c.domain do
  writef("%2i", c(i,j).locale.id);
writeln();
```

3 Jacobi and Gauss-Seidel

Read and understand the Jacobi iteration program, `jacobi.chpl`. Now write a Gauss-Seidel iteration program, using only a single array to hold data. (*Hint:* The lecture notes contains the core section of code for this program.)

4 File I/O

1. Read and understand the demo program `fileIO.chpl`. The program reads and writes regular 4-byte integers. Compile and run it. Now modify the program to read and write single-byte integers. (*Hint:* With the modified program, there should be 64 integers in the `input` file.)
2. (Optional) Write a coarse-grained parallel version of file I/O program, `fileIO2.chpl`. In this program, create a worker routine to be run on each locale:

   ```
   for loc in Locales do
     on loc do worker();
   ```

   The worker routines run concurrently. Each worker routine declares a local array of size `N/numLocales` (where `N` is the integer count of the input file); it then opens a channel to the input file, and reads a proper section from the file to the array. View the input file as having `numLocales` equal-sized sections; since locale id starts from 0, the worker routine on locale `k` should read the `k+1`-th section.

   Afterwards, the worker routine performs similar actions for the output — opens a channel to the output file and writes its array to the corresponding section in the file.

Submission

Write a short report summarizing your work. Submit it with your programs through the "Lab 9" folder on Canvas. The submission deadline is the end of tomorrow (Friday).