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: PGAS Programming in Chapel

1 Matrix Multiplication

The file `mmul.chpl` contains a matrix multiplication program in Chapel. Convert the program to a PGAS version, `mmul-dm.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();
```

The file `domainmap.chpl` contains a demo program for domain maps. You may use it as a reference.

2 Jacobi Iteration

The file `jacobi.chpl` contains a shared-memory version the Jacobi iteration program. Read and understand it, then convert it to a PGAS version, `jacobi-dm.chpl`. Add a similar code block as above to verify that array `a` is indeed partitioned across the locales.

3 Gauss-Seidel Iteration

Now write a share-memory version of the Gauss-Seidel iteration program, `gauss-seidel.chpl`. The program should use only a single array to hold data. Note that this week's lecture showed the core section of code for this program. You need to think of a new way to track convergence. The approach used in the Jacobi iteration, of comparing values of the old and new arrays, is no longer valid since there is only one array. (*Hint:* Comparing the old and new values of a single scalar variable is allowed.)

Compile and run this program. Do you observe a faster convergence rate compared to the Jacobi iteration program?

[**Optional**] This default version allows race conditions, which means the program is nondeterministic. But it always converges. In this week's lecture, we also showed two alternative deterministic versions: Red-black and Wavefront. Choose one to implement.

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).