

CS510 Languages and Low Level Programming: Portfolio submission, Topic 10

Due on June 3, 2016 at 11:59pm

Mark P. Jones Spring 2016

Konstantin Macarenco

Topic 10. Use practical case studies to evaluate and compare language design proposals.

With the lack of specific language for Low Level Programming, I picked two languages with Parallel Programming in mind (General C + MPI library, and Chapel - new domain specific language for parallel programming created by Cray). Parallel programming complexity is similar, if not greater than LLP, with many potential issues on the top of regular mistakes there are Concurrency Issues such as race conditions, and deadlocks. MPI - message passing interface is a well known standard for external parallel computing. Chapel - is build upon C/MPI is much simpler - it is a modern high level language that hides all the intricacies of Message Passing. Chapel syntax is similar to Python, with many alike features. Chapel uses C/MPI as an intermediate layer, i.e. first it compiles to C.

I Compare implementation of Jacobi-Laplace algorithm in C/MPI vs Chapel.

Jacobi-Laplace is a simple approach for solving Laplace equation with $O(n^3)$ complexity, used in many scientific applications. As $n \rightarrow \infty$ performance is greatly reduced, hence the desire to run it in parallel mode.

Laplace equation : $\phi_{i,j}^{t+1} = \frac{1}{4}(\phi_{i+1,j}^t + \phi_{i-1,j}^t + \phi_{i,j+1}^t + \phi_{i,j-1}^t), 0 < i, j < N$, i.e. current cell in a matrix is equal to a quarter of the sum of it's neighboring cells.

External im

1. Simple C/MPI implementation:

- Complexity - implementing it in C/MPI requires matrix partition,
- Size - ≈ 620 *lines* with comments

2. Simple Chapel implementation:

- Complexity -
- Size - ≈ 60 *lines* with comments