LCSModuleDescription.pdf

This module can be used in a variety of ways depending on the purpose of the class. In a parallel computing class this module can be used as a stand alone project, where the instructor uses class time to explain the problem and describe its solution and then asks the students to implement the solution. In a scientific computing class this module could be used as an example of parallel computing applied to Genomic analysis. Additionally, there are many different activities an instructor may want to have the students do, which can vary in the amount of student work. The activities could range from simply downloading and running the code to fully developing the entire application. Students could also be provided the main function and asked to implement either one or all of the table filling functions. The students could also be asked to write the function that matches the strings given the cost table has already been filled. It all depends on the purpose of the class, perhaps graduate courses would have higher expectations than undergraduate. Some options are detailed below.

1. The instructor could use the 25 minute class to explain the LCS problem and how a Dynamic Programming solution is feasible because it is an optimization problem, there are overlapping subproblems, and the optimal solution to a smaller instance of the problem is used in the final optimal solution. The instructor could then provide a serial solution (filling the table one row or column at a time). Once the students understand the serial solution they can learn about the data dependency problem, which prevents the serial solution from simply being made parallel by adding some OpenMP directives. The data dependency problem is that not only is each cell along a row dependent on the immediately preceding cell being previously computers, the entire row depends on the previous row being previously computed. Therefore, cell computation is not independent of other cells. One way to remove some of the cell dependence is to compute the value of the cells along diagonal lines. This way, all the cells along the diagonal are independent of each other even though there remains a dependence between the diagonals. At this point the instructor could ask the students to implement the diagonal table filling approach or they could ask the students to parallelize the serial, diagonal filling algorithm.

2. Another assignment could be to have the students simply download, compile and run the code with different sets of Genomic data files, where plots of run time versus text and pattern length and the number of threads are made. The data files could be augmented so the number of threads would be the first integer in the file or the number of threads could be a second command line argument.

3. Another assignment would be to ask the students to implement the matching algorithm, given that the table has been filled.

4. Download and run the code on Blue Waters with both of the supplied (or additional) data files. How does the performance change when running the program as the number of threads vary?

5. Add a function to fill the table by columns and compare its performance with the function that fills the table by rows.