Questions:

1. (20 points) Consider solving a linear system of the form $Ax = b$, where $A$ is an $n \times n$ symmetric, tridiagonal matrix. Specify pseudocode for the parallel cyclic reduction algorithm specialized to this particular class of matrix. It is OK to make the same assumption we made in class in regards to the relationship between $p$ and $n$. **Be sure to pay particular attention to minimization of the storage of $A$.**

2. (20 points) Consider solving a linear system of the form $Ax = b$, where $A$ is an $n \times n$ symmetric matrix which is factored as $A = LDL^T$. Here $L$ is an $n \times n$ unit lower triangular matrix and $D$ is an $n \times n$ diagonal matrix. Design and implement in C/C++ an efficient linear solver for symmetric linear systems based on an $LDL^T$ factorization of $A$. Demonstrate that your code obtains correct results on a small linear system with a known solution.

Generate timing results from the solution of a symmetric linear system with a $500 \times 500$ left-hand side matrix. In addition, generate timing results from the solution of a symmetric linear system with a $1000 \times 1000$ left-hand side matrix. **You must run your code on the slurm cluster as demonstrated in class.**

Submit the following items electronically via e-mail to shontz@ku.edu: (1) your C/C++ code and (2) your PBS script.

In addition, submit the following items in hard copy: (1) your C/C++ code, (2) your PBS script, (3) output demonstrating your code works (as indicated above), and (4) output containing your timing results.

Both the electronic and hard copy submissions are required for this question.

3. (20 points) Determine the computational complexity of the symmetric linear solver you developed in Question 2 based on the use of an $LDL^T$ factorization.

In addition, answer the following question: Do your timing results from Question 2 support your computational complexity in Question 3? Why or why not?

4. (20 points) Specify pseudocode for a parallel symmetric linear solver based on an $LDL^T$ factorization.