PROJECT PROPOSAL

JOHN J. TRAN

1. Background

In general, we seek to solve the linear equations $Ax = b$ where A is a sparse coefficient matrix. The frontal and multifrontal methods compute the LU factors of A, where its factorizations are tiered dense sub-matrices (or frontal matrices). These frontal matrices are recursively assembled and pivot operations are done on them until the L and U factorizations are completed. For a symmetric sparse matrix, minimum degree ordering can be applied to reduce the non-zero fill-ins and the ordering can be generated with a symbolic computational tree. On the other hand, for unsymmetric sparse configurations, the tree is best represented by an acyclic graph.

2. Problem Description

Up to now, the various approaches for constructing the supernodal elimination tree are well understood and explored in the three parallel computing paradigms: (1) message passing, (2) multithreaded, and (3) a hybrid of the two aforementioned approaches. Most recently, several attempts to further exploit highly parallel computing paradigm, *vis-à-vis* Graphics Processing Unit (GPU), have bear promising results. One such approach is to employ GPUs as inexpensive accelerators to factor the large supernodes. In this approach, the factoring panels or frontal matrices are constructed on the host system and work is farmed to the GPU. With this approach, only a subset of computing intensive sub-matrices are GPU-tasked because the transfer cost of data block has been shown to be extremely expensive.

3. Proposed Work

In light of the preceding observation, one can (and should) ask: can we not task the GPU to do the scheduling and directly orchestrating and communicating with the dense compute kernels on the frontal nodes? This effectively reduces the extremely expensive GPU/CPU communication bottleneck. For this project, we propose to look at various ways in which the GPU can conduct multifrontal factorization and scheduling of the dense compute kernels.