Components to Support Parallel Adaptive Unstrcutured Mesh Simulations on Massively Parallel Computers

M.S. Shephard[a], C. Smith[a], S. Seol[a], D. Ibanez[a] and A. Ovcharenko[a]

[a] *Scientific Computation Research Center, Rensselaer Polytechnic Institute, USA*

It is well known that adaptive methods are the most effective means to obtain reliable solutions and control the amount of computation required. However, for many problems the best adaptive method still requires a level of computation that demands massively parallel computing. The evolving nature of adaptive computations constantly changes the computational balance, thus requiring structures to control parallel operations and communications as well as efficient dynamic load balancing. The first part of this presentation will briefly overview FMDB, a parallel distributed mesh infrastructure and associated partition model for unstructured mesh calculations. After that consideration will be given to a set of procedures to effectively control the large numbers of small messages associated with parallel mesh adaptation and to quickly load balance directly using information available through mesh topological adjacencies. The last part of the presentation will discuss extensions to this infrastructure to effectively support newer generations of massively parallel computers that have a large number of cores per node.

A general-purpose communication package, IPComMan, supports parallel applications that rely on computation characterized by large number of messages of various sizes, often small, that are focused within processor neighborhoods. IPComMan provides a utility for dynamic applications based on explicit consideration of the neighborhood communication pattern, use of non-blocking MPI functions, and message packing to asynchronously manage message flow control and reduce the number and time of communication calls. Results on IBM Blue Gene/P and Cray XE6 computers show that the use of neighborhood-based communication control leads to scalable results when executing generally imbalanced mesh adaptation runs.

Parallel unstructured simulations at extreme scale require that the mesh be distributed across a large number of processors with equal workload and minimum inter-part communications. The goal of ParMA is to dynamically partition unstructured meshes directly using the mesh adjacency information to account for multiple criteria. Results will demonstrate the ability of ParMA to dynamically rebalance large meshes on large core count machines accounting for multiple criteria.

As the number of cores per node increases, the use of standard message passing on the node will introduce parallel inefficiencies. Thus it becomes important to take advantage of threaded parallel operations and node level shared memories. However, hand threading of some unstructured mesh operations, such as adaptive mesh modification, would be extremely complex and, because of the highly conditional nature of the operations, difficult to make scalable. An alternative approach is to introduce a two level decomposition of the mesh, first to the node level and then to the core level, where internode parallel operations are through standard message passing and intranode paralleization over the node's cores is via threading. The goal of this approach is for FMDB and IPComMan follow-on to control the message passing and threading processes. The IPComMan follow-on, PCU (Parallel Control Utility) will add architecture awareness, threading and improved buffer management. In this way the unstructured mesh applications can continue to either operate as they currently do, or to be modified to account for the two level partitioning, which should be much easier to do than hand threading the application. The presentation will indicate our current status on attempting to implement such an approach.

See see http://www.scorec.rpi.edu/software.php for more information on these and other unstructured mesh software components.