ESTO Computational Technologies Project

PYRAMID: Parallel Unstructured Adaptive Mesh Refinement Library

Modern... Simple... Efficient... Scalable...

Technology Description

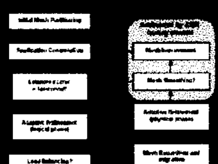
An advanced software library supporting parallel unstructured adaptive mesh refinement for large-scale scientific and engineering simulations.

State-of-the-Art Design

- Efficient object-oriented design in Fortran 90/95 with MPI for message passing.
- Automatic mesh quality control, dynamic load balancing, mesh migration, partitioning, integrated mathematics and data structure management routines, all in parallel.
- Scalable to hundreds of processors and millions of elements using triangular (2-dim) and tetrahedral (3-dim) elements.
- Power, completeness, and ease of use.

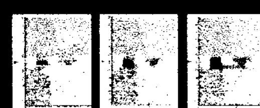
Parallel Adaptive Process

Initially the (generally random) input mesh must be repartitioned and redistributed after loading from disk. The application calculation and local error-estimation steps occur followed by a logical AMR process that decides how new elements will be created.



Load balancing techniques, if required, will determine the best location for the coarse elements by using algorithms to minimize data movement and boundary communication.

A physical refinement stage will create new elements from the coarse elements in the proper load balanced location with quality control.



Quality control maintains element geometry.

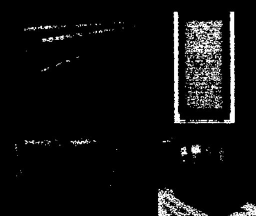
Multi-Million Element Adaptive Refinement

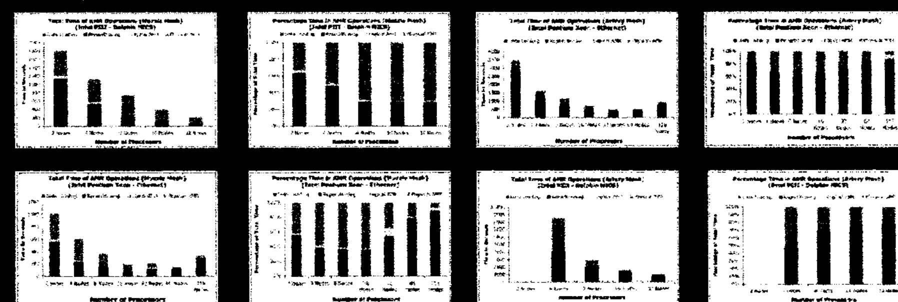
Colors show processors that contain elements that are load balanced across partitions on the parallel machine.

Performance Modeling on Clusters

Significant computational power is required for finite element analysis of large models so we use clusters, often with high speed networks, for our software development.

Pyramid runs on massively parallel computers as well as clusters.





A Minimal Pyramid Program

Users provide the initial mesh and per-element error indicator and Pyramid handles all of the parallelism issues.

```
PROGRAM pyramid_example
USE pyramid_module
implicit none
  type (mesh), dimension(2) :: meshes
  call PAMR_INIT()
  call PAMR_LOAD_MESH_PARALLEL( meshes(1), in_file )
  call PAMR_REPARTITION( meshes(1) )
   do i = 1, [illegible]
      call PAMR_ERROR_EST( meshes(1), meshes(2) )
      call PAMR_LOGICAL_AMR( meshes(1) )
      call PAMR_REPARTITION( meshes(1) )
      call PAMR_PHYSICAL_AMR( meshes(1), meshes(2) )
   end do
  call PAMR_ELEMENT_COUNT( meshes(2) )
  call PAMR_VISUALIZE( meshes(2), [illegible] )
  call PAMR_FINALIZE( mpi_active = .true. )
END PROGRAM pyramid_example
```

Current and Future Collaborations

Partnering with others influences library development.

Charles D. Norton, John Z. Lou, and Thomas A. Cwik

Earth Science Data Systems, Applied Cluster Computing Technologies