Domain Decomposition

Aim : Minimize communications cost by distributing the cells in the domain in some optimal way.

$$\text{time} = \text{latency} + \frac{\text{message size}}{\text{communications bandwidth}}$$

Approach #1

- Ideal for large problems, slow networks or whenever many small messages are passed.
- Neglects latency.
- Total length of inter-domain boundaries is minimized (which is length of all shared data)..in other words minimizes shared part of perimeter/surface area of the domain.
- Requires a-priori knowledge of number of domains (compute nodes) and hence has to be done online every time the problem is run on a different number of nodes.
- Popular techniques need eigenvalue of the connectivity matrix and thus can be expensive for large problems.
- However, produces best grids for large problems.

Approach #2

- Ideal for small problems, fast networks or whenever few big messages are passed.
- Neglects second term as communication bandwidth is assumed large.
- The total number of messages exchanged is minimized.
- Since the above minimization corresponds to minimization of bandwidth of the cell connectivity matrix, this approach is independent of the number of domains (compute nodes).
- A large knowledge base exists in this area and is readily accessible so implementation is sound and easy.

These domain decomposition techniques are absolutely essential since we are dealing with unstructured grids in which the ordering of grid cells is generally arbitrary (especially, if say, Advancing Front Method was used to create the 2D/3D grids or Delaunay Triangulation for the 2D grids).

Approach #2 is used in PUMA as it is aimed at machines with high communications bandwidth.