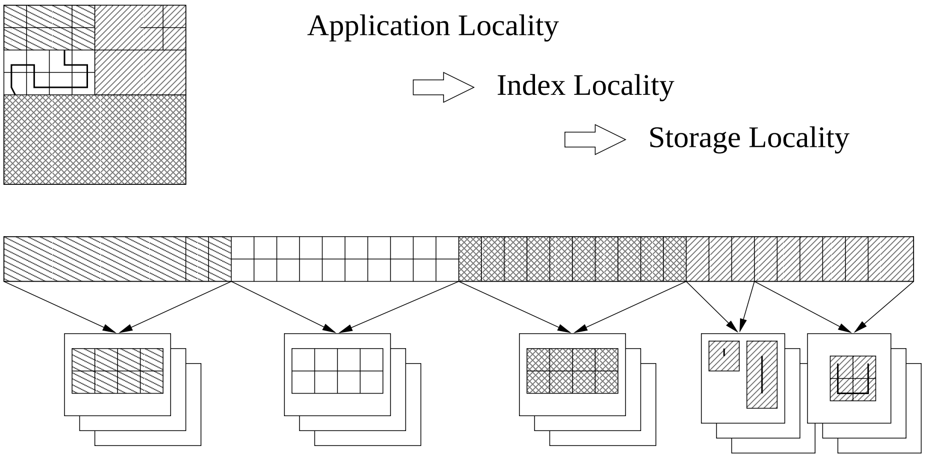

Figure 7:

Distributed Dynamic Storage - Partitioning and Communication

An instance of a DAGH is mapped to an instance of the HDDA. The granularity of the storage blocks is system dependent and attempts to balance the computation-communication ratio for each block. Each block in the list is assigned a cost corresponding to its computational load. In case of an AMR scheme, computational load is determined by the number of grid elements contained in the block and the level of the block in the AMR grid hierarchy. The former defines the cost of an update operation on the block while the latter defines the frequency of updates relative to the base grid of the hierarchy. Note that in the representation described above, space-filling mappings are applied to grid blocks instead of individual grid elements. The shape of a grid block and its location within the original grid is uniquely encoded into its space-filling index, thereby allowing the block to be completely described by a single index. Partitioning a DAGH across processing elements using this representation consists of appropriately partitioning the DAGH key list so as to balance the total cost at each processor. Since space-filling curve mappings preserve spatial locality, the resulting distribution is comparable to traditional block distributions in terms of communication overheads.

The individual objects referenced by each key, usually arrays, are stored in the standard layouts used by C and/or FORTRAN so that computational routines written in conventional sequential languages can be applied to these stored objects.