WHAT IS CLAIMED IS:

1. A multiprocessor system comprising:

a plurality of nodes and at least one memory,

wherein each node contains at least one processor (CPU), at least one cache private to the node, and at least one cache location buffer (CLB) private to the node,

wherein each CLB stores location information values, each location information value indicating a location associated with a respective data unit,

wherein each location information value stored in a given CLB indicates the location to be either a location within the private cache disposed in the same node as the given CLB, to be a location in one of the other nodes or to be a location in memory,

wherein the values of the data units are kept coherent by a cache coherence protocol and the cache coherence protocol updates the location information values in accordance with movements of their respective data units.

2. The multiprocessor system of claim 1, further comprising:

at least one global cache which does not reside in any of the nodes and wherein the location information value can also identify a location within the at least one global cache.

3. The multiprocessor system of either claim 1, wherein the coherence protocol classifies a continuous region of data units to be a private region if the data units only have their respective location information values stored in one or more CLBs which are private to one and the same node.

4. The multiprocessor system of claim 1, further comprising at least one global directory storing location information about the data units, wherein the global directory is used to implement the coherence protocol and wherein the nodes can spill and fill location information to and from the global directory.

5. The multiprocessor system of claim 1, wherein the continuous region has associated therewith information about which nodes have location information for the data units associated with the continuous region stored in a CLB private to the node.

6. The multiprocessor system of claim 5, wherein only nodes indicated to have location information for the data units associated with the continuous region stored in a CLB private to the node can have copies of the associated data unites stored in their respective private caches.

7. The multiprocessor system of claim 4 wherein handling a global invalidation request for a data unit sent from a requesting node to the directory involves the steps of:

performing a lookup in the directory to find information associated with the data unit and its region;

invalidating copies of the data unit in nodes, excluding the requesting node, indicated to possibly have copies in their private caches; and

updating location information in the private CLBs associated with nodes indicated to have location information for the data unit.

8. The multiprocessor system of claim 4, wherein handling an eviction of a data unit from a node to a new location involves the steps of:

copying the value of the data unit to the new location; and

updating the data unit's location information in the private CLBs associated with nodes storing location information for the data unit .

9. The multiprocessor system of claim 4, wherein the directory implements a blocking function that blocks certain coherence requests from being serviced by the directory.

10. The multiprocessor system of claim 4, wherein the private CLBs and the directory, all storing location information, form a separate inclusive cache hierarchy and wherein the local and global caches, storing the data units, form a non-inclusive hierarchy.

11. A multiprocessor system comprising:

a plurality of nodes, more than one global cache and at least one memory,

wherein each node contains at least one processor (CPU) and at least one cache which is private to the node,

wherein the values of the data units stored in the caches are kept coherent by a cache coherence protocol performing lookups in a directory storing locality information for the data units,

wherein the location information identifies a location where the data unit resides as being one of: a node, a global cache or memory.

12. The multiprocessor system of claim 11, where the location information identifying a global cache, also identifies a location within that cache as a location for the data unit.

13. The multiprocessor system of claim 11, further comprising:

a plurality of global caches, in each of which more than one node can allocate space for its requested data units, wherein one or many nodes have one of the plurality of global caches placed closer to them than other global caches and wherein mapping a data unit placement to a global cache is done independently of the data unit's address.

14. The multiprocessor system of claim 11, further comprising replacement counters associated with the global caches which are used to determine in which global cache a data unit should be allocated.

15. The multiprocessor system of claim 11, wherein the global caches can be configured to have all nodes access one common copy of a data unit stored in one global cache or where it can be dynamically decided to create an additional replicated copy of a data unit in a specific global cache.

16. The multiprocessor system of claim 11, wherein a continuous region of data units has an address tag associated with their corresponding location information in the directory, and the address tag identifies the region.

17. A method comprising:

storing data units in private caches in nodes of a multiprocessor system, wherein each node contains at least one processor (CPU), at least one cache private to the node and at least one cache location buffer (CLB) private to the node;

storing, in each CLB, location information values, each location information value indicating a location associated with a respective data unit, wherein each location information value stored in a given CLB indicates the location to be either; location within the private cache disposed in the same node as the given CLB, to be a location in one of the other nodes, or to be a location in a main memory;

maintaining coherence of values of the data units using a cache coherence protocol; and

updating, by the cache coherence protocol, the location information values stored in the CLBs in accordance with movements of their respective data units.

18. The method of claim 17, further comprising:

storing data units in at least one global cache which does not reside in any of the nodes and wherein the location information value can also identify a location within the at least one global cache.

19. The method of claim 17, wherein the coherence protocol classifies a continuous region of data units to be a private region if the data units only have their respective

location information values stored in one or more CLBs which are private to one and the same node.

20. The method of claim 17, further comprising:

storing location information about the data units in at least one global directory, wherein the global directory is used to implement the coherence protocol and wherein the nodes can spill and fill location information to and from the global directory.