3.16 MRAPI Potential Future Extensions

With the goals of implementing MRAPI efficiently, the APIs are kept simple with potential for adding more functionality on top of MRAPI later. Some specific areas include read/copy/update (RCU) locks, non-owner remote memory allocation, application-level metadata, locking of resource lists, and informational functions for debugging, statistics (optimization) and status. These areas are strong candidates for future extensions and they are briefly described in the following subsections.

3.16.1 RCU (read/copy/update) locks

Although this feature is common in certain SMP operating systems, it is not clear that the feature scales well to embedded and/or non-SMP contexts. If research currently underway at various universities dispels this concern then RCU locks may be a feature worth adding to MRAPI.

3.16.2 Non-owner remote memory allocation of remote memory

Certain use cases considered by the working group indicated the usefulness of giving a node the ability to obtain memory from a different node. After consideration the working group determined that the API could be kept simple and this ability could be satisfied by using MCAPI messaging to allow one node to ask the other node to allocate on its behalf. In the future if this proves to be too inefficient for real-world application scenarios then this feature could be considered.

3.16.3 Application-level metadata

It is clear that application-level metadata can be used for rich higher-level functionality. The MRAPI working group believes this should be a layered service which can be built using a combination of MCAPI and MRAPI features. In case this proves to be difficult in the future we may wish to consider adding this feature.

3.16.4 Locking of resource lists

While similar APIs for resource management do provide functions for locking lists of resources, for now we believe this can be done well enough with mutexes and semaphores, especially given that MRAPI cannot enforce such locks (being a cooperative sharing API). If in the future it is proven we were mistaken this could be a feature we could consider adding.

3.16.5 Debug, Statistics and Status functions

Support functions providing information for debugging, optimization and system status are useful in most systems. This is worth future consideration and would be a valuable addition to MRAPI.

3.16.6 Multiple Semaphore Lock Requests

It may be useful to add a feature that allows allocation of multiple counts of semaphore at once, instead of recursively calling the lock().

3.16.7 Node Lists for Remote Memory Creation Routines

We may wish to add a node list parameter to the shared memory creation routines. This would provide symmetry with the shared memory routines.