```
        /* Wait for previous write operation to complete */
        if(!first)
        {
              mrapi_wait(&request[cur_buf], &status, NO_TIMEOUT);
              if (status != MRAPI_SUCCESS) {
                ERR("Unable to complete remote memory write for DMA");
              }
        }

  }

  /* Wait for final write operation to complete */
  mrapi_wait(&request[1-cur_buf], &status, NO_TIMEOUT);

  // CHECK STATUS FOR ERROR
  if (status != MRAPI_SUCCESS) {
        ERR("Unable to complete final remote memory write for DMA");
  }
  /* Detach from remote memories */

  mrapi_rmem_detach(sw_cache_hndl, &status);

  // CHECK STATUS FOR ERROR - DETAILS OMITTED
  if (status != MRAPI_SUCCESS) {
        ERR("Unable to detach from remote memory sw cache");
  }

  mrapi_rmem_detach(dma_hndl, &status);

  // CHECK STATUS FOR ERROR - DETAILS OMITTED
  if (status != MRAPI_SUCCESS) {
        ERR("Unable to detach from remote memory DMA");
  }

  /* Notify Node 1 that we are done */
  notify_node1();

  return 0;
  }
```

Remote Memory Use case 2

This use case captures the scenario where one processing node has a very small amount of RAM, and requires RAM on another processing node to be made available to store intermediate results. This is