```
        ERR("Unable to create remote memory for sw cache");
      }
      /* Send Node 2, as integers, values of the pointers for
'entities_to_be_processed' and 'START_OF_HEAP' */
      send_to_node2((int)entities_to_be_processed);
      send_to_node2((int)START_OF_HEAP);

      /* Promote 'scores_to_be_computed' to allow remote access via DMA */
      dma_hndl = mrapi_rmem_create(AGREED_ID_FOR_DMA,
                                   scores_to_be_computed,
                                   MRAPI_ACCESS_TYPE_DMA,
                                   NULL,
                                   number_of_entities*sizeof(float),
                                   &status);

      // CHECK STATUS FOR ERROR
      if (status != MRAPI_SUCCESS) {
          ERR("Unable to create remote memory for DMA");
      }
      /* Node 2 can now find these remote memory buffers, and work with them
*/

      /* Node 1 waits until Node 2 has finished (using some appropriate
mechanism) */
      wait_for_notification_from_node2();

      mrapi_rmem_detach(sw_cache_hndl, &status);

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

      mrapi_rmem_delete(sw_cache_hndl, &status);

      // CHECK STATUS FOR ERROR
      if (status != MRAPI_SUCCESS) {
        ERR("Unable to delete remote memory for sw cache");
      }

      mrapi_rmem_detach(dma_hndl, &status);

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

      mrapi_rmem_delete(dma_hndl, &status);

      // CHECK STATUS FOR ERROR
      if (status != MRAPI_SUCCESS) {
        ERR("Unable to delete remote memory for DMA");
```