```
int cur_buf = 0; // Selects which buffer we are currently using.

do {
      unsigned int offset_for_next_entity =
       address_of_next_entity_to_process - start_of_node1_heap;

      /* Read an entity from Node 1's memory, via software cache.
      We use a blocking operation because we need the result to
      continue processing, and we hope the cache will mean that the
      result is held locally and will thus arrive quickly. */
      mrapi_rmem_read(sw_cache_hndl,
                              offset_for_next_entity,
                              &next_entity_to_process,
                              0,
                              sizeof(Entity),
                              1, /* num_strides is 1 */
                              0, /* rmem_stride is irrelevant */
                              0, /* local_stride is irrelevant */
                              &status);

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

      result_buffers[cur_buf][num_entities_processed % BUFFER_SIZE] =
            process( & next_entity_to_process );
      num_entities_processed++;

      address_of_next_entity_to_process =
            (unsigned int)(next_entity_to_process.next);

      if((num_entities_processed % BUFFER_SIZE) == 0)
      {

            // CHECK STATUS FOR ERROR - DETAILS OMITTED

            /* Issue non-blocking DMA of buffer-full of results back
            to Node 1's memory.  We use a non-blocking operation
            because we do not need to wait for the write to
            complete in order to continue processing the list: it is
            preferrable to overlap communication with computation. */
            mrapi_rmem_write_i(
                        dma_hndl,
                        num_entities_processed*sizeof(float),
                        result_buffers[cur_buf],
                        0,
                        BUFFER_SIZE*sizeof(float),
                        1, /* num_strides is 1 */
                        0, /* rmem_stride is irrelevant */
                        0, /* local_stride is irrelevant */
```