```
  // This function takes our large allocation and fills in the hash
  // table.  In particular, it takes the large memory region and
  // splits into a flow_hash_table_t object and a free list
  // containing MAX_FLOWS flow_info_t objects.
  flow_table = init_flow_table(mem, MAX_FLOWS);

  // Initialize locks for each bucket, and for the free list.
  for (int i = 0; i < HASH_BUCKETS; i++)
  {
    flow_table->buckets[i] = mrapi_mutex_create(i, NULL, &status);
    if (status != MRAPI_SUCCESS)
      die("Couldn't allocate mutex for bucket.\n");
  }
  flow_table->free_list_lock =
    mrapi_mutex_create(HASH_BUCKETS + 1, NULL, &status);
  if (status != MRAPI_SUCCESS)
    die("Couldn't allocate mutex for free list.\n");
}

// This could be implemented using MCAPI.
barrier();

if (rank != 0)
{
  mrapi_shmem_hndl_t mem_hndl = mrapi_shmem_get(MY_SHMEM_ID, &status);
  if (status != MRAPI_SUCCESS)
    die("mrapi_shmem_get() failed.\n");

  flow_table = (flow_hash_table_t*) mrapi_shmem_attach(mem_hndl, &status);
  if (status != MRAPI_SUCCESS)
```