```
  if (mrapi_status  != MRAPI_SUCCESS) { WRONG }

  /* Get the cache attributes */
  filter = MRAPI_RSRC_CACHE;
  root = mrapi_resources_get(filter, &mrapi_status);
  l3cache = root->children[0];
  uint32_t cache_hits;
  mrapi_resource_get_attribute(l3cache, 1, (void *)&cache_hits,
sizeof(cache_hits), &mrapi_status);
  if (mrapi_status != MRAPI_ERR_RSRC_NOTSTARTED) { WRONG }


  /* Start the L3 cache hit monitoring */
  mrapi_dynamic_attribute_start(l3cache, 1, &l3cache_hits_rollover,
&mrapi_status);
  if (mrapi_status != MRAPI_SUCCESS) { WRONG }

  while (rollover == MRAPI_FALSE) {
    mrapi_resource_get_attribute(l3cache, 1, (void *)&cache_hits, attr_size,
&mrapi_status);
    if (mrapi_status != MRAPI_SUCCESS) { WRONG }
    printf ("cache hits = %d",cache_hits);
  }

  /* stop the L3 cache hit monitoring */
  mrapi_dynamic_attribute_stop(l3cache, 1, &mrapi_status);
  if (mrapi_status != MRAPI_SUCCESS) { WRONG }
  mrapi_resource_get_attribute(l3cache, 1, (void *)&cache_hits, attr_size,
&mrapi_status);
  if (mrapi_status != MRAPI_ERR_RSRC_NOTSTARTED) { WRONG }

  /* finalize */
  mrapi_finalize(&mrapi_status);
  if (mrapi_status != MRAPI_SUCCESS) { WRONG }
```

6.6.2 mrapi_resource_get() examples

Below are a series of metadata use cases based on a single system. The use case figures are graphical representations of the resource data structure returned by a call to `mrapi_resources_get()`.

Consider as an example two CPUs and two memories connected by two busses, with a node running on each CPU.