

The purposes of the L4 refactoring exercise are:

1. Handle the synchronization of the CPU & Xeon Phi
2. Make maximum use of both CPU and Xeon Phi
3. Hide (as much as possible) the mechanical difficulties
From the tracking/analysis code.

We will effectively replace the `L4Interface::handleEvent()` Call
With 3 calls: `init_proc()`, `simu_proc()` & `decide()`.

There will be a strict order for the function calls in each event:

1. `init_proc(rdr, buffer)` is called first
2. `simu_proc(rdr, buffer, n)` are called after `init_proc()` is finished. There could be several such functions all called at the same time.
3. `decide(rdr, buffer)` is called after ALL `simu_proc()` Functions have finished.

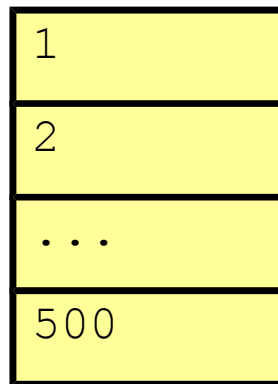
The `l4Interface` class functions will be declared static, so they can not use any non-local variables. The reason is that the same `L4Interface` class will not be used for The various calls.

My suggestion would be to create an "L4EventData" structure With all variables that need to be used in different calls, and store it in the "initial_result" region of the buffer

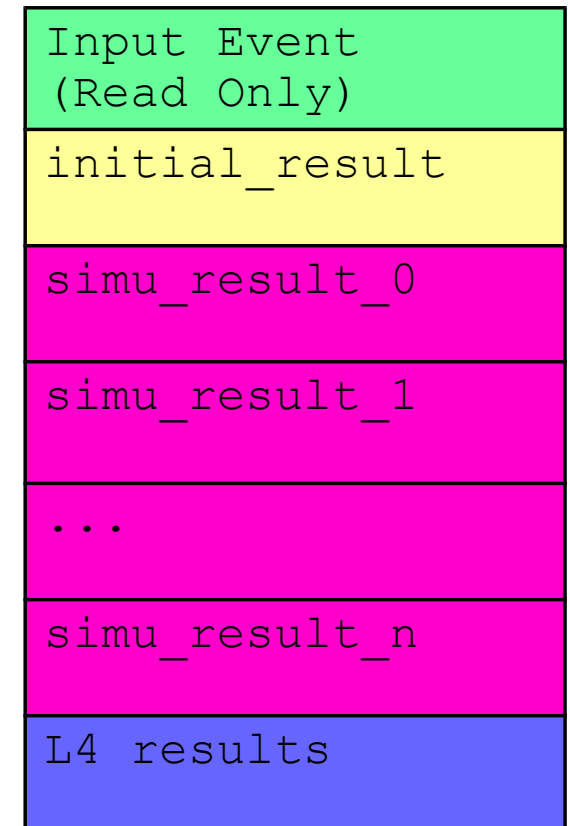
The buffer structure needs to be somewhat more complex:

- * L4Task keeps buffers for 500 events
- * Each of the functions has its own fixed memory within its buffer
- * A function can only read from prior regions to be filled
(The `simu_proc()` functions can not read each others memory)
- * A function can only write to its own memory.
- * The L4Results can keep the same "hlt_blob" structure as before
But I assume that these pointers may point back to previous Results...

Event Buffers:



Each Buffer:



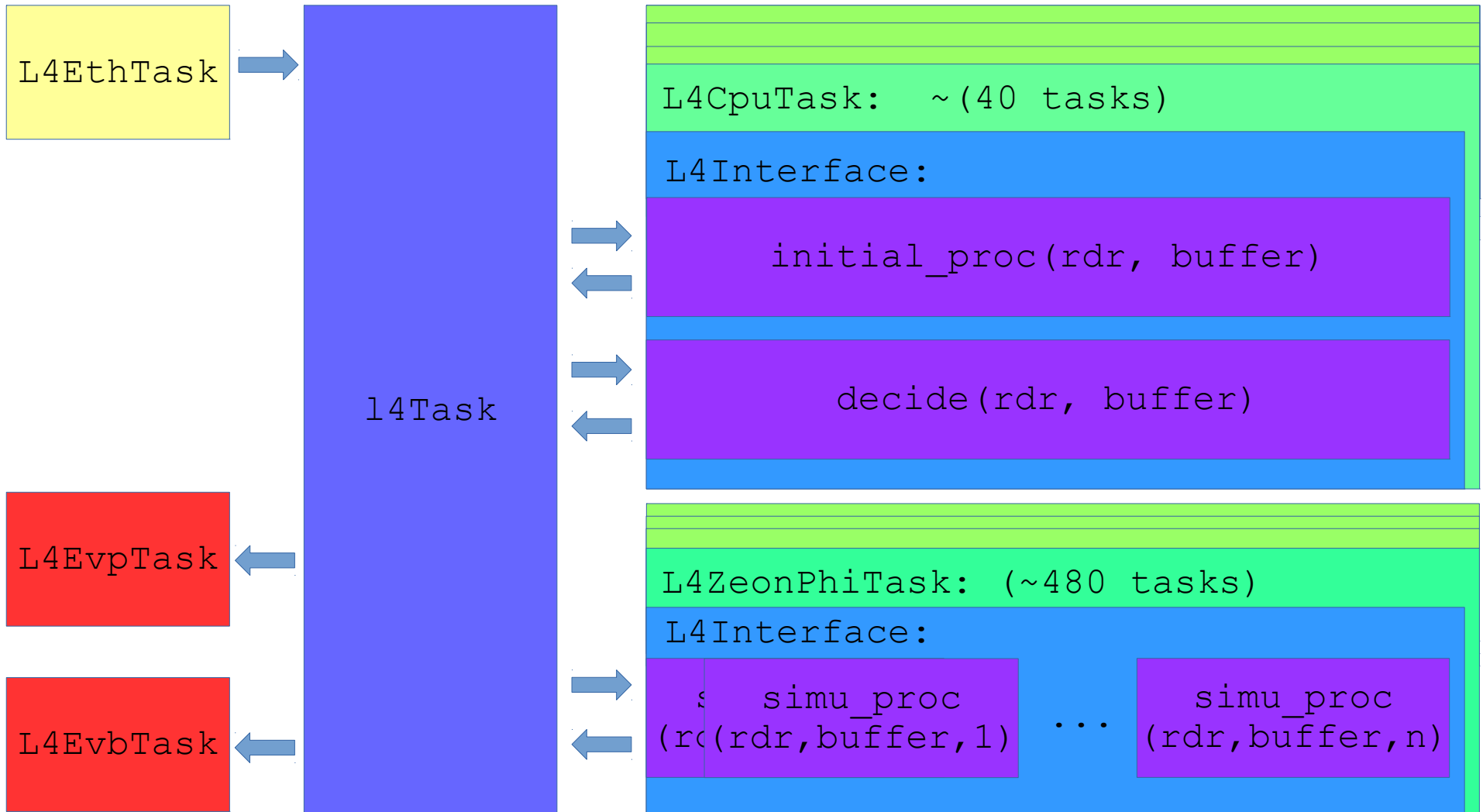
Assume:

- * 500 Buffers
 - * Input Event (30MB)
 - * CPU Stage 1 (6MB?)
 - * xeon_phi results (6MB?)
 - * xeon_phi "n" (1?)
 - * L4 results (MAX 6MB)
- * Total size: 24GB

L4 system structure:

The overall structure is the same, except:

1. L4TriggerTask is split into CpuTask & ZeonPhiTask
2. `handle_event(rdr, *output_blobs)` is replaced by 3 stages
`initial_proc()`, `simu_proc()` & `decide()`



Allocation of buffers for the xeon phi nodes:

- * It's not yet clear what, if any, specific criteria are required.
- * My plan is to add a function `init()` to the L4 interface which will be called when the `L4TriggerTask` is launched. This should allow the setup and storage of the `xeon_phi` control data and/or sockets to be handled. This information is tied to the thread not to any given event which will potentially be farmed out to multiple threads.
- * If the xeon phi memory needs to be tied to specific events (rather than copied into the buffers as I've specified) Then we will need to discuss the requirements more and modify the buffer structure and/or the buffer allocation Scheme.