Meeting of the Architecture Working Group with F.Wickens
27th September

Summary by Saul Gonzalez

Main topics discussed or raised during the meeting:

1. Difference between LVL2 farm and sub-farm
2. Issues with LVL2 configuration
3. Monitoring
4. Pseudo-ROS
5. Event queuing in LVL2
6. PESA request for RoI data
7. Offline dependencies
8. Thread- vs. process-switching

Summary:

The discussion started with trying to pin down the definition of a sub-farm. The concept of a sub-farm is not yet fully developed, but Fred listed three possibilities: (1) The collection of processors assigned to a single L2Supervisor; (2) The collection of processors attached to peripheral switches (in turn attached to a central switch); and (3) the collection of processors under a single online control leaf. It is natural to group a fixed number of processors to a single L2SV because the L2SV performs the load balancing (otherwise the various L2SVs would have to talk to each other).

It may also be possible to have more than one sub-farm attached to a single L2SV. The present model is that, within a sub-farm, all processing units have the same configuration. This still needs to be discussed. Fred also mentioned that this was not essential for the TDR.

There was confusion with (3) above, but Fred explained that (3) was motivated by the problem of configuration, when large amounts of data need to be uploaded to the processors via the online services. Many MB of meta-data are needed by algorithms at startup and it is not clear if the online (and the control network) can handle this load. Currently all configuration data is lost when a worker thread is terminated – this could be avoided by local caching in a database. This issue will probably not be addressed for the TDR. However, we still need to get an order of magnitude estimate of the size/frequency of this meta-data (task for PESA, detectors).

The question was raised of who/where was gathering trigger statistics to account for all events and fragments going through the system. In addition to the standard monitoring tasks running at each processor, the "request/response" nature of the system allows this type of integrity check (if no response to a request within a timeout → problem). The conclusion was that we need more use cases for error conditions, including hard faults and event processing faults.

Fred mentioned that the "request/response" rule is broken for the pseudo-ROS, which could be addressed with a "send/acknowledge". During EB, the pseudo-ROS is

treated like a normal ROS. Presently the pseudo-ROS is responsibility of the Data Collection group. The pseudo-ROS is important (but not essential) for EF seeding and for LVL2 diagnostics.

Fred explained in detail the LVL2 selection, stressing that the HLT LVL2 selection made extensive use of the Data Collection libraries. The 'input thread' receives messages from the L2SV (accumulating events is a queue) and from the ROS (locally buffering until requested fragments arrive). Worker threads in the L2PU grab events from the queue and process them by calling the PESA Steering Controller (the top HLT component) with the L1 result. As algorithms process the event, they request data through the "Data Manager", one of the HLT's interfaces to the Data Flow. From the discussion that ensued, it was not clear why worker threads couldn't handle their own data requests, instead of using a single input queue. This point needs to be clarified.

The design of the PESA component that requests RoI data is based on "detector elements", which is an offline construct. This means that the atomic unit of data requests is one ROB/request, which is unacceptable due to the system overheads. Our offline collaborators say that this situation will be corrected in the future, but probably not before the TDR. The status is that we have a version of the software without this flaw, but not conforming to the PESA design document ("London Scheme").

Fred stressed that this problem was a symptom of our dependencies in the offline software. As originally conceived, the LVL2 environment was not the same as the EF/offline. Since L2/EF/offline now have the same environment, we get the long-term benefits with some short-term pain. We depend on EDM, StoreGate, and Identifiable Containers. We also have a dependency on LHCb for Gaudi, which we have modified for thread-safety.

There was a discussion on the merits of thread switching vs. process switching. The conclusion was that, since the DC software can run in either mode, multi-thread vs. multi-process performance measurements should part of the upcoming testbed program.