Table of Contents

Introduction

Collection of requirements for DataflowEvolution project.

Requirements

Lists of Use Cases (UC) and User Requirements (UR).

Sysmon

Dedicated twiki page: requirements

HLTSV

Use cases

- **UC_SV10** HLTSV receives L1-Result from RoIB
  - Or, the sw RoIB component running inside the HLTSV receives data from n S-links and assembles a L1-Result fragment
- **UC_SV20** HLTSV delivers a L1-Result to a DCM running on one of the available HLT nodes
- **UC_SV30** HLTSV receives a clear message from DCM when the latter has either fully collected an event or has rejected it
- **UC_SV31** HLTSV re-assigns an event to another DCM in case of s/w or h/w problems on the initial node
- **UC_SV40** HLTSV groups and broadcasts `ROBs-clear` messages to ROS PCs.
- **UC_SV50** HLTSV receives a message from DCM when a processing slot (a cpu core) is available
  - N.B.: during HLT processing after full data collection, the fragments can be deleted from ROBs, but a cpu core is still busy in EF processing

User requirements

- **UR_SV01: HLT input rate**
  The HLT system *shall* be able to sustain an input rate of 100kHz
- **UR_SV02: Single HLTSV**
  A single HLTSV instance *should* be able to receive L1-Results at 100kHz and distribute them to the HLT processing nodes
  - If the requirement cannot be satisfied, the HLT farm should be divided in sub-farms, each managed by its own HLTSV
- **UR_SV03: HLT farm balancing**
  The HLTSV *shall* take care of the HLT farm balancing
- **UR_SV04: Software based RoIB**
  It *should* be possible to merge the HLTSV with a software based RoIB. More... Close
  - This scenario is possible only with a single HLTSV
  - The combined system *shall* be able to receive, assemble and distribute L1-Results at 100 kHz
- **UR_SV10 Condition data injection**
  The HLTSV *should* be an injection point for condition data changes. More... Close
  - Presently updates for conditions data at lumi block boundaries are communicated to the L2Pus and EFPTs by additional fields in the CTP ROB fragment. Updates include prescale

changes and some conditions data folders for detectors. Changing these fields in the CTP fragment requires a firmware update and has strong space limitations.

- **UR_SV20: Event ownership**
  The HLTSV is in charge of an event (L1ID) until it is rejected by the HLT or fully collected by the DCM

- **UR_SV21: Fault tolerance**
  The HLTSV *shall* be able to identify h/w or s/w problems in an HLT node and re-assign the event allocated to the given node to another machine in the farm. More... Close
  - For example via the implementation of an event time-out mechanism

- **UR_SV22: Event reassignment**
  Reassigned events *shall* be properly labeled to prevent data duplication. More... Close
  - The node owning the initial copy of the event could be still alive and it could still accept the event

- **UR_SV40: ROB occupancy load**
  The HLTSV *shall* inform the ROSs when the fragments of a given event can be deleted.
  - I.e. when an event has been rejected or fully collected by an HLT node.

- **UR_SV41: ROS load**
  The HLTSV *should* reduce as much as possible the load on the ROS PC.
  - In order to reduce the ROS PC load, the HLTSV *should* group the `ROB-clear` messages. The grouping *shall* not contribute significantly to the ROB occupancy.

- **UR_SV50: Occupancy and deadtime**
  The HLTSV *shall* provide information about its buffers occupancy and dead-time

DCM

Use cases

- **UC_DCM01** The DCM registers itself to the HLTSV, providing the number of available processing slots. More... Close
  - "Processing slots" == "Number of HLPUs on the node"

- **UC_DCM10** The DCM receives an L1-Result (list of ROB fragments) from HLTSV

- **UC_DCM20** The DCM assigns an L1-Result (list of ROB fragments) to a free HLTPU

- **UC_DCM30** On HLTPU request, the DCM collects ROB fragments and makes them available to the caller HLTPU

- **UC_DCM40** When an event is either rejected or fully collected, the DCM send a clear message to the HLTSV: the fragments of that event can be deleted from ROBs

- **UC_DCM50** The DCM forwards accepted event to the SFO

- **UC_DCM60** The DCM force accept events in case of HLTPU processing problems.

User requirements

- **UR_DCM01: Single DCM instance per node**
  There *should* be a single DCM instance for each HLT node.

- **UR_DCM04: Input**
  The DCM *shall* be able to receive L1 results from the HLTSV. The data *shall* be formatted as array of `eformat::ROBFragment`

- **UR_DCM06: Data collection**
  The DCM *shall* be able to collect data from the ROS PCs according to a list of ROB identifiers provided by the HLTPU. The data *shall* be formatted as array of `eformat::ROBFragment`

- **UR_DCM08: Event assembly**
  The DCM *shall* assemble accepted events as `eformat::FullEventFragment` according to the list of ROB identifiers provided by the HLTPU.
    - ♦ An empty list could indicate all the collected ROBs

- **UR_DCM10: Output**
  The DCM *shall* send assembled events to the SFO as `eformat::FullEventFragment`

- **UR_DCM12: Emulation**
  For testing purpose, it *shall* be possible to internally emulate input, output and data collection functionalities

- **UR_DCM14: ROB mapping**
  The mapping of ROBs to ROSes *should* reside in the dataflow software. More... Close
  In the HLT, it *should* not be used further than internally in the HLT `ROBDataProviderSvcROBDataProvider`. Algorithms should only work with ROBs and their mapping to geometrical regions. This is already the case now and should be conserved to assure maximum decoupling of dataflow (OKS) configuration and HLT configuration.

- **UR_DCM16: Data size**
  The DCM *shall* be able to manage events with sizes ranging from few kB to tens of MB. More... Close
    - ♦ The average event size can change run by run
    - ♦ In each run physics and calibration events are mixed:
        - ◊ Physics event: about 1-2 MB
        - ◊ Calibration events: from few kB to tens of MB

- **UR_DCM18: PUIO**
  The DCM *shall* implement the `PUIO` to provide, on demand, ROB fragments to the HLTPU

- **UR_DCM20: HLT result**
  The DCM *shall* be able to receive an HLT result fragment (an `eformat::ROBFragment`) via the `PUIO` interface and it *shall* append the fragment to the accepted event

- **UR_DCM24: Data integrity**
  The DCM *shall* guarantee data integrity inside the node

- **UR_DCM26: Fault tolerance for HLTPU problems**
  The DCM *shall* be able to identify processing problems in an HLTPU (application crashes or processing dead-loops) and force accept the involved event. More... Close
    - ♦ ◊ In the current system (EFD),
        - · a PU crash is identified via the "socket hangup" caught by the unix domain socket server

· a processing dead-loop via an event processing timeout

- **UR_DCM28: Fault tolerance for DCM crashes**
  In case of DCM crash, it *should* be possible to recover fully-collected events More... Close
  - For example: at restart the DCM *should* be able to retrieve data from disk
  - NB: event not yet built are still hosted in the ROS system and they will be assigned to other nodes by the HLTSV(s)

- **UR_DCM30: ROS occupancy**
  The DCM *shall* reduce as much as possible its contribution to the ROBs occupancy.
  - The DCM shall inform the HLTSV as soon as an event has been fully collected or it has been rejected

- **UR_DCM31: processing power exploitation**
  The DCM *shall* exploit as much as possible the available processing power.
  - The DCM *should* inform the HLTSV when an HLTPU has done with a given event and therefore a processing slot is free. More... Close
    - NB: the DCM shall inform the HLTSV as soon as an event has been has been rejected or fully assembled, but in the latter case the event is still being processed (EF) and a core is busy.
    - A dedicated message should be used to avoid the assignment of events to busy nodes.

- **UR_DCM40: Data compression**
  The DCM *shall* provide online raw event compression. More... Close
  - Simplify the offline treatment of the raw data and reduce by a factor ~2 the bandwidth and storage requirements for the SFOs.
  - Strategies have to be defined to accommodate:
    - the SFO processing of the data (e.g. uncompressed event header, event header duplication, ..)
    - the stripping of calibration events (e.g. selective compression, event duplication in the EF, ..).

- **UR_DCM41: Data sampling**
  The DCM *shall* provide event sampling capability. The DCM *should* provide the possibility to sample also events rejected by the HTLPU.

- **UR_DCM42: High rate calibration streams**
  For calibration purpose, the DCM or the HLTPU *should* provide a mechanism to manage high rate of small events. More... Close
  - ToDo: create dedicated page with muon calibration requirements provided by Enrico Pasqualucci
  - In the current system muon tracks are selected in each L2PU, collected via a hierarchy of dedicated servers and written to disk in a specific stream. The data bypasses SFI, EFD and SFO. See Calibration_stream_software.pptx
  - The current muon calibration infrastructure could in principle be re-used with minimal effort in the evolution.
  - On the other hand in the evolution the muon calibration information will be one step away from the storage system. It makes therefore sense to try to route these data through the standard data-flow, with the aim of reducing the number of components to be maintained.
    - Based on the experience with the muon calibration in the current system, in particular in terms of rates and bandwidth, dedicated aggregation and transport strategies have to be developed.
  - In case a very high rate of calibration data is expected at the SFOs, one should consider both the implications on the SFOs performance and the operational impact on monitoring

requirements.

- **UR_DCM50: Occupancy and deadtime**
  The DCM *shall* provide information about its buffers occupancy and dead-time

- **UR_DCM70: Full event pre-fetching**
  The DCM *should* have the possibility to trigger full data collection independently from HLT request. More... Close
  - E.g.: the DF could automatically initiate full data collection (pre-fetching) if already ~ X% of all ROBs are retrieved.
  - N.B.: This requirement is under discussion

- **UR_DCM80: ROS pre-fetching**
  The DCM *should* have the possibility to collect all the ROBs in a give ROS independently from HLT request. More... Close
  - E.g.: If a request to a ROS is issued which wants to retrieve already most of the ROBs in this ROS, DCM may retrieve then already the complete set ROBs in this ROS.
  - N.B.: This requirement is under discussion

HLTPU

Use cases

- **UC_PU00** The HLTPU subscribes to the DCM running in its node.

- **UC_PU10** The HLTPU receives a L1-Result (as array of `eformat::RobFragment`) from DCM via a dedicated interface (`PUIO`)

- **UC_PU11** The HLTPU forwards the L1-Result to the HLT steering layer via a dedicated interface (`hltinterface`)

- **UC_PU20** The HLT steering sends ROB requests to the HLTPU

- **UC_PU21** The HLTPU forwards the requests to the DCM and returns the collected data to the HLT steering

- **UC_PU30** The HLT steering provides an event selection decision

- **UC_PU40** In case of accepted event, the HLT steering can provide an HLT result fragment (as `eformat::RobFragment`) to be appended to the event

User requirements

- **UR_PU00: HLT interface**
  The HLTPU *shall* host the HLT steering framework. The interface between HLTPU and HLT steering *shall* be defined by a dedicated interface (`hltInterface`)

- **UR_PU02: PUIO interface**
  On demand, the HLTPU *shall* provide the HLT steering framework with ROB fragments (array of `eformat::RobFragments`). The data is obtained from the DCM via the `PUIO` interface.

- *UR_PU03: PUIO interface: emulation *
  For testing purpose it *shall* be possible to emulate the DCM functionality More... Close
  - For example using an implementation that reads fragments from a file

- **UR_PU03: PUIO interface: data**
  The HLTPU *shall* receives data from DCM as array of `eformat::RobFragments`.

- **UR_PU10: Number of HLTPU**
  There *shall* not be limit on the number of HLTPUs running on each node
    - ♦ The value depends on the number of available cores and on the amount of memory

- **UR_PU20: Data protection**
  The HLTPU *shall* not have any possibility to corrupt the ROB fragments.

SFO

At the moment, no changes for this component are foreseen.

Message passing

- **UR_MP01 Network latencies**
  It *shall* be possible to measure network latencies

- **UR_MP?? .....**

Comments

Major updates:
-- AndreaNegri - 19-Mar-2012

%RESPONSIBLE% AndreaNegri
%REVIEW% **Never reviewed**

This topic: Sandbox > JustForTest
Topic revision: r4 - 2013-01-29 - AndreaNegri