ALICE DAQ Internal Note

02 April 2002

Pierre.Vande.Vyvre@cern.ch

Summary of first meeting on rare triggers (02 April 2002)

Present: T. Anticic, C. Fabjan, L. Musa, K. Safarik, J. Schukraft, P.Vande Vyvre

---

1. Introduction

1.1. Do we really have a problem with rare triggers?

The result of the first batch of simulation done by T. Anticic has shown potential problems with the way rare triggers are handled. The problem arises inevitably whenever the data volume sent over the DDL's (as given by the trigger mix and event size) exceeds the data volume that can be processed by the DAQ/HLT system. In this case, the buffers in the intermediate stages of the DAQ (GDC's, LDC's, or RORC's) get full and by the action of the back-pressure this is propagated up to the detectors that become busy almost all the time. As soon as the busy is released, the frequent triggers are obviously the most likely to fire and to put the detector into busy state again.

1.2. Requirements for a solutions

Two requirements must be met to solve the problem:

- Keep the detector lifetime as high as possible in order to catch as many rare triggers as possible;
- Keep the backend part of the DAQ system (event building and mass storage) as busy as possible, i.e. as close as possible to the maximum installed bandwidth.

2. Implementation

A note circulated by Pedja and proposing a feedback mechanism from the DAQ to the CTP has been used as input for the discussion.

2.1.1. Downscaling

The downscaling currently foreseen in the CTP will allow limiting the rate of each trigger class. It would be possible to adjust the downscaling so that the adequate rate of frequent triggers is obtained.

There are however two main drawbacks:

- During a fill of the LHC, the luminosity will decrease by a factor of between 2 and 4) and it will therefore be required to adjust the downscaling factors during the fill.

- It will be needed to know the history of the changes of the downscaling factors for the normalisation during reconstruction and analysis.

2.1.2. Feedback mechanism from DAQ to CTP

The solution consists of making a time-sharing of the detector lifetime between periods dedicated to frequent triggers and rare triggers and periods reserved to rare triggers only.

The solution is based on the Class Mask bits in the CTP. Each trigger class has a corresponding Class Mask bit that must be enabled for the corresponding class to fire. Two thresholds would be defined for the LDC's buffers: a high-level threshold or almost-full and a low-level threshold corresponding to almost-empty.

When the almost-full threshold is reached, the Class Mask bits corresponding to frequent triggers are disabled and the system is reserved for rare triggers. Given the total amount of buffering in the system, the DAQ back-end will be kept busy for a while.

When the almost-empty threshold is reached, the Class Mask bits corresponding to frequent triggers are enabled and the system is open to all triggers. The occupancy of DAQ buffers will increase again.

Data-taking periods will then be time-shared between two types of behaviour. When all types of trigger are enabled, the LDC's buffer occupancy will increase rapidly until reaches the almost-full threshold is reached. This will then be followed by a period reserved to rare triggers. The LDC's buffer occupancy will then decrease because the rare triggers do not saturate the DAQ bandwidth.

One can estimate that the PC's available at startup time will include several Gbytes of memory. If one assumes that 1 Gbyte is dedicated to the data buffering, one TPC LDC is able buffer of the order of 3000 events (70 MBytes collected by 216 DDL's). Even at the maximum acquisition rate of 200 Hz, the time constant is of the order of 15 seconds. In these conditions, the solution proposed would not require a separate low latency network to carry the feedback messages.

A big advantage of the proposed solution is also that it is based on existing hardware in the CTP.

2.1.3. Discard data in the LDC's

It would be possible to discard the data in the LDC's just after the transfer over the DDL's. However, this seems a waste of lifetime and bandwidth. It would also not be a big simplification over the previous scheme as it would require a synchronisation between all the LDC's to decide which events to discard.

3. Next Steps

The existing simulation program will be modified to include the proposed implementation. Different scenarios will be simulated to verify the behaviour.