Summary of Meeting on ST Threshold Settings

Date: 10/12/2007
Present: J. Luisier ,M. Knecht , M. Needham. K. Rinnert

Aims:

Start to understand how to to set-up the chain needed to download the thresholds to the Tell1 boards for the ST. Since the VELO are more advanced in this area Kurt was invited to share his experience.

What needs to be done

We need automatically to download the clustering thresholds to the Tell1 board. Kurt sketched how this is forseen to be done (Fig. 1). All communication goes via PVSS. This ensures things are done in a consistant manner. The procedure is roughly as follows. A Gaudi job will write the Tell1 constants into a file. This is then picked up by PVSS and written to the configuration database. When a run is started PVSS reads back these numbers and downloads them to the TELL1 boards. The numbers are then read back from the TELL1 boards and stored in the Conditions database. Since the last step will involve XML files Kurt suggests that communication between the Gaudi job and PVSS should go via XML files as well. This seems a reasonable idea.

The numbers that need to be set are:

- Header correction parameters. It needs to be understand at what granularity these need to be set.

- Pedestals. Per strip value to seed the start of the automatic pedestal following. Since this value will be updated automatically having a precise initial guess is not necessary.

- Pedestal mask. One value per strip. Dead strips are masked off.

- Hit threshold for CMS algorithm. The lowest granularity is per strip.

- Strip Threshold. The lowest level of granularity forseen is per strip.

- Confirmation threshold. The lowest level of granularity forseen is per Beetle.

- Spillover threshold. The lowest level of granularity forseen is per Beetle.
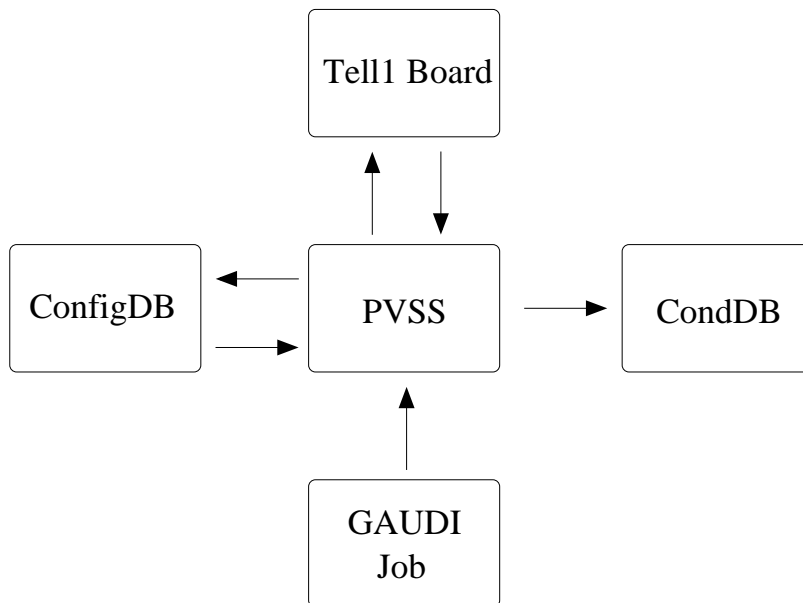
Figure 1: Downloading of Tell1 constants.

Typically the thresholds will be multiples of the noise. In addition, the requirement to have high efficiency and a low noise cluster rate means that reasonable first values of many of the parameters can be determined using pedestal data alone. The only exception is the spillover threshold which needs data to be determined. First guesses of the thresholds might be:

CMS Threshold $S/N = 5$

Strip threshold $S/N = 3$

Confirmation threshold $S/N = 4$

High threshold $S/N = 10$

Code needs to be developed in Gaudi to make the noise per strip. The starting point should be the output of the emulation code. To reduce the amount of information to a manageable level averages per module should also be made. It should be investigated if this code can be integrated into the Online monitoring code.

An open issue is the granuality. At the start we would like to have a minimal number of thresholds across the detector. Over time, if needed, we should

envisage to add more detail. We could do this in two ways. First, the output of the Gaudi job could be an XML file with the highest possible granularity. In that case no further modification of the format is needed if more detail is added. However, many numbers are stored and downloaded to PVSS that are all the same. The other option is to store only the needed parameters in the database. Only at the step of reading from the configuration database to PVSS (or writing from PVSS to the conditions) database is the list of constants expanded (or contracted) to the granularity of the Tell1 boards. The VELO have chosen to use the first option.

Kurt has started to try to develop tools that deal with the interaction of the PVSS and the conditions database. Hopefully this work can be adapted for the ST.

Actions:

- Johan (helped by Mathias and Anne) will start to make an algorithm to look at the mean and rms of the noise per strip (and also summary numbers for each module). He will start from the output of the emulation code of Anne and use the first data taken at the pit.

- Matt and Kurt will try to understand how the interaction with the configuration database proceeds.

- Kurt will make public his scripts that deal with the interaction between PVSS and the conditions database [and also conditions and the cfg files]. These may be useful for the start of ST development.

- Matt will make a proposal for the format of the XML files. We should try to make this as similar as possible to the VELO.

- Mathias will start to look at the online monitoring aspects.