Online Data Monitoring in the LHCb experiment

Data monitoring is an important concept to verify the detector performance with quick response in case of problems (hardware aspects) and to verify the performance of the software based event filtering and rejection (software aspects). The hardware infrastructure of LHCb allows monitoring data taken by the experiment in the readout boards, the high level trigger farm and in a dedicated monitoring farm.

Monitoring requirements:
- Must not interfere with data taking activities.
- The monitoring applications are heterogeneous, they may execute different code depending on the functionality.
- Require different types of events as input data.
- May run in several instances, if one single executing application is not able to gather sufficient statistics.

The FPGA boards on the TELL1 process the data from the frontend electronics. Basic monitoring based on registers read from the FPGAs is possible using a "Credit Card" PC mezzanine (Intel 486-Based 32-bit AMD processor, SLC Linux). The information is published using the concepts described below.

Collecting Monitoring Information
The main output of monitoring applications are histograms and scalars like counters. Typically identical tasks execute in several instances; hence, information of the same type has to be collected and summed before being published saved and analyzed. The information may be collected in several stages:
- Processes [Monitoring & Event filter tasks] publish the histogram information.
- Adders collect the histogram information, add the histograms and publish them. If many sources must be collected a tree-like hierarchy of Adders may be formed to sum the information.
- Saver tasks read the added histograms from the Adders, save them to ROOT files and notify the Analysis tasks.
- Analysis tasks are activated after saving the histograms which check distributions, compare against reference histograms and raise alarms in the event of irregularities.
- The Histogram Presenter can connect while data taking to the adder tasks to display interactively sets of histograms. The inventory in the histogram database allows storing and retrieving predefined views.

Implementation
Most tasks producing monitoring information execute in the Gaudi data processing environment and publish the information using a monitoring component specialized for the online environment. The information is published with the DIM protocol.

Control Infrastructure
All tasks producing monitoring information are controlled by the experiment control system (ECS) using PVSS – like any other component of the experiment. They are represented by a common finite state diagram and are configured coherently.

Data processing applications are grouped in a tree structure and are manipulated by the RunControl component. At the highest level the operator issues commands to the RunControl, which is responsible for the overall orchestration of the state transitions.
- The top level is represented by the DataFlow component with children representing data processing sub-systems.
- Subsystems are the HLT farm, the Monitoring and the Storage system.
- At the lowest level each subsystem consists of commodity processors executing processes which produce monitoring information.

Hardware
~50 standard Intel based dual CPU dual core processors.

Data Transport to the Monitoring Farm
- The events accepted by the HLT trigger are sent to the storage system.
- From the storage system events are sent to the monitoring farm in 2 steps:
  - First to redistribution or relay box(es). To minimize destructive interferences with data taking and to reduce the load on the storage system data are only sent on request.
  - The data are forwarded to the monitoring farm nodes and analyzed.

Partitioning of the Monitoring Farm
- Several DAQ partitions share the monitoring farm.
- Sharing requires a flexible allocation mechanism based on the number of monitoring tasks to be executed by each partition. Each monitoring task may require a different type of events.

Realization of Partitioning
- Define a structure of logical slots in the monitoring nodes.
- During configuration each partition allocates the slots required and a monitoring task is assigned to each slot. This defines the data streams which need to be routed from the storage system to the monitoring nodes.
- Instrument the storage system with tasks sending the data to the relay box.
- Instrument the relay box with tasks receiving the data and forwarding them to the correct monitoring nodes.

Implementation Choices
All task implementations are based on the LHCb Gaudi framework. To simplify and decouple the transport and to access the data, a buffer manager concept is used. Producer tasks declare data to a shared memory based buffer manager. When data are available Consumer tasks get notified and access them. 3 types of processes execute in the system:
- Sender tasks receive events from the local buffer manager and send them to the Receiver tasks using TCP/IP. The receivers declare the received data to the local buffer manager.
- Monitoring tasks receive events from the local buffer manager and analyze them and create monitoring information in the form of histograms.