DATABASE PARA O COSMIC

Marco Leite

Formato para os eventos (preliminar)

```
#==========================================================================================
class Event :

    def __init__ (self):
        #super(Event,self).__init__() #micropython does not like multiple inheritance
        """
            Just creating this class will instantiate the containers. Not everything will be filled by the detector firmware.
            We may have some type of service to prepare some extra type of information (like events per cosmic block etc.)


            Event : The  event record equivalent. For consistency, RunNumber and Dtector Id, while redundant, will be sent for
                    consistency. This concept should (and will) evolve as we have a DB in place and more experience with the
                    system.
        """

        self.Event = {
            "Format"      : None ,
            "RunNumber"    : None ,
            "DetectorId"  : None ,
            "EventNumber"  : None ,
            "CosmicBlock" : None ,
            "Timestamp"   : None ,
            "TriggerBits" : None ,
            "TDC" : {
                "0" : {
                    "Data" : [None, None, None, None, None],
                    "Calib": [None, None]
                } ,
                "1" : {
                    "Data" : [None, None, None, None, None],
                    "Calib": [None, None]
                }
            } ,
            "ClockCount" : None ,
            "TrigerCounter" : None
        }

#==========================================================================================
```

DATABASE PARA O COSMIC

Formato para os eventos (preliminar)

ver o arquivo dump.dat na agenda


```
{"Epoch": 4009558015, "CounterClocks": 2764593408, "DetectorId": 0, "RunNumber": 12345,
"TriggerCounter": null, "EventCounter": 3121872896, "ClockCount": null, "Format": 1,
"EventNumber": 4917, "TDC": {"1": {"Calib": [null, null], "Data": 5112064, "Calib1":
3521728, "Calib2": 12407809}, "0": {"Calib": [null, null], "Data": 6554112, "Calib1":
15075584, "Calib2": 15360512}}, "Timestamp": 622234535, "TriggerBits": 71, "CosmicBlock":
2}
{"Epoch": 2751135742, "CounterClocks": 2031504640, "DetectorId": 0, "RunNumber": 12345,
"TriggerCounter": null, "EventCounter": 3138650112, "ClockCount": null, "Format": 1,
"EventNumber": 4918, "TDC": {"1": {"Calib": [null, null], "Data": 6684928, "Calib1":
3456192, "Calib2": 13259777}, "0": {"Calib": [null, null], "Data": 6422784, "Calib1":
14944512, "Calib2": 15491584}}, "Timestamp": 622234536, "TriggerBits": 16, "CosmicBlock":
2}
{"Epoch": 4244504575, "CounterClocks": 47700992, "DetectorId": 0, "RunNumber": 12345,
"TriggerCounter": null, "EventCounter": 3155427328, "ClockCount": null, "Format": 1,
"EventNumber": 4919, "TDC": {"1": {"Calib": [null, null], "Data": 6684928, "Calib1":
3456192, "Calib2": 13259777}, "0": {"Calib": [null, null], "Data": 1573120, "Calib1":
14682368, "Calib2": 16146944}}, "Timestamp": 622234536, "TriggerBits": 71, "CosmicBlock":
2}
```


Cada linha
um evento

O que precisa ser definido :

- Database (MySQL ?)
- Schema (Baseado no event format)
- Host (AWS ?)
- Daemon (python com ORM ?)
- ?

Proposta: Levantar um DB server para conectar ao broker do AWS e receber os eventos via MQTT para começar