Real-time Environment in the DAQ PCs

Tonko Ljubicic/BNL
at the Forward Upgrade Software Meeting
Feb 12, 2021
V1.0

Topics

- What are the so-called DAQ PCs and what do they do
- Software environment and “rules” to abide to
- FCS setup
- STGC setup
- FST setup
- Summary

DAQ PCs Hardware

- destination of all the detector's data coming from the various frontend electronics boards and hardware interfaces on the STAR detector in the WAH
- majority of interconnect interfaces are optical fibers
  - DAQ PCs house fiber receiver cards, different types, all custom built PCI cards
- most DAQ PCs reside in racks in the "DAQ Room"
  - visible through the window of the "Control Room"
- Each detector uses 1 or more (up to e.g. 36 for TPX) DAQ PCs
  - which we often call "Sectors" for historical reasons
- Each DAQ PC houses 1 or more (up to e.g. 8 for FCS) fiber interfaces
  - which we often call "RDOs" for historical reasons
- DAQ PCs are connected to STAR's Event Builders which assemble final events to tape (HPSS)
  - usually single 1x GbE interfaces, very rarely 2x1 GbE ("bonded") interfaces
  - limits the throughput-to-tape to 100 MB/s per DAQ PC!
- DAQ PC specs vary (mostly depending on the year of purchase) but typically 8+ core Intel Xeons, 2-3 GHz, 4-32 GB memory, 1-2x GbE copper interfaces to EVBs, small disk for e.g. pedestals and some amount of cached data

DAQ PCs Software

- they all run the same Linux, currently SL7
  - all boot diskless off of “daqman” so installing new packages is not straightforward but possible
  - code development/compilation is done on daqman and the binaries are executed by the DAQ PCs
- all run some flavor of custom code which reads in data, listens to Run Control, processes the data, packs the data into DAQ formats, sends the data to EVBs for further storage, etc.
- all of this is done under pseudo (“soft”) real-time conditions!
  - malloc/free is frowned upon (and so is new/delete)
  - processing time and (more importantly) stability is critical
    - if the data gets corrupted we can only hope that it will get noticed downstream (e.g. EventPool)
  - no messing around with random “freeware” -- all code is custom C++/C
  - NO “root”
    - however, I am prepared to discuss various root classes which are known to be stable
    - BTW, the usual “root” thinks of itself as an operating systems and grabs many system resources -- not allowed!
- yours truly is responsible for all this

Typical Data Processing in DAQ PCs

- data compression tasks
  - pedestal subtraction, zero suppression
  - cluster finding (TPC)
  - without these forms of data compression STAR would not be able to reach our XX billion event goals ⇒ this compression is essential!
- data coherency/quality checks
  - checks the hardware and “alarms”/”complains” in case of issues
    - informs the Shifcrew via DAQ Monitoring messages
    - can force a Run Stop if the problem is deemed CRITICAL
    - sends email to people who request to be notified of certain conditions
- helpers
  - pedestal calculations online, bad channel determination, gain checks, etc.
- this is performed on every event, all the time, automatically
  - vs e.g. the Event Pool or Fast Offline which doesn’t check *all* events
- caveat is that a DAQ PCs doesn’t see the entire detector’s worth of data but only its corresponding section/sector/chunk/fibers
  - we can use HLT for full-detector and/or full-event (all detectors) analysis

FCS Specifics

- 11 DAQ PCs, each with 8 fibers connected to DEP boards (digitizers and trigger processors)
  - 16 CPU cores, 16 GB RAM, 500 GB disk, 1 GbE interface to STAR EVBs
  - IP addresses: fcs01, fcs02,... fcs11
- processing
  - pedestal subtraction, zero suppression
  - “bit checker” software checks the coherency of all trigger bits on each and every event

STGC Specifics

- 4 DAQ PCs, each reads 4 fibers connected to ROB boards
  - 16 CPU cores, 16 GB memory, small disk, 1 GbE interface
  - IP addresses stgc01… stgc04
- Processing
  - unknown to me at this time, depends on the VMM format but I assume we will need some form of compression e.g. noise rejection at least
  - note that each DAQ PC's fibers will be connected to the frontend in such a way that it gets all of the data from 1 STGC plane
    - one could do per-plane analysis followed by data compression: peak finders, etc
  - other ideas?

FST Specifics

- 1 (old) PC, 6 fibers connected to ARC boards
  - 8 CPU cores, 4 GB RAM, small disk, 2x GbE interfaces were used for IST(?)
  - IP address fst01
  - we could use another PC if deemed necessary (fst02)
    - or even more (there are a number of old PCs available)
- processing (assuming IST)
  - pedestal subtraction, zero suppression
    - very important since the raw data volume is too large for “tape”!
  - we could also try to do some sort of noise rejection?
  - perhaps also get rid of all APV’s timebins and sum up to 1 value?
  - other thoughts?

Summary

- DAQ PCs are there and reasonably powerful
  - they run Linux, like all Offline/Online machines
    - simple to code (but with caveats → see earlier slides)
  - they process/"see" every event
- Data-compression in some form is required!
- also note that the more you do on these PCs the less you need to do Offline
  - witness the TPC cluster-finding