WP4
Mu2e Software updates

R.Donghia, LNF-INFN

MUSE Scientific Board Meeting

December 20, 2019

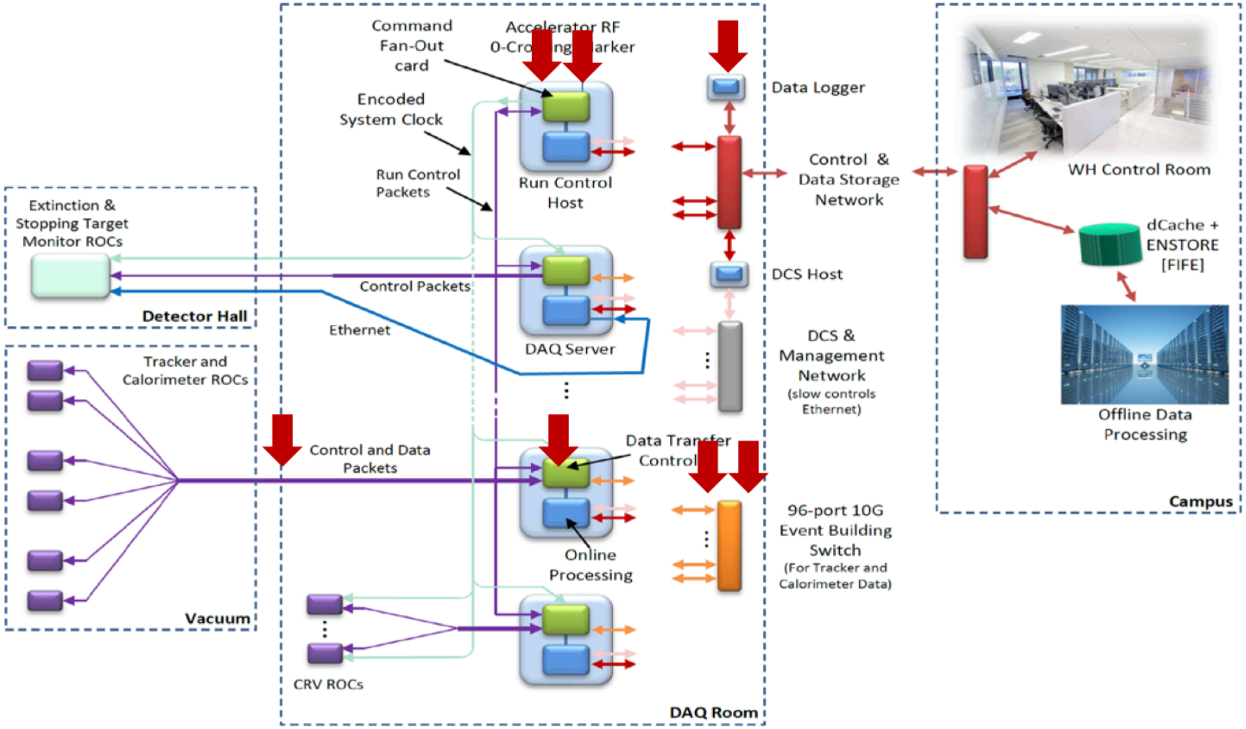
Outline

- TDAQ risks
- Upgrading the FastCalo modules in Offline

TDAQ risk

- TDAQ is central for acquiring data
  → It is important anticipate systems failure
- Risk separeted in 2 categories:
  - **TDAQ hardware** (physical hardware, from ROC to buffer disk…). The risk is that some components fail, needs to be replaced and a new spare must be bought
  - **TDAQ performance**, related to the performance of the trg system. The risk is not having enough processing power to keep up with rate, despite the software optimization

TDAQ hardware

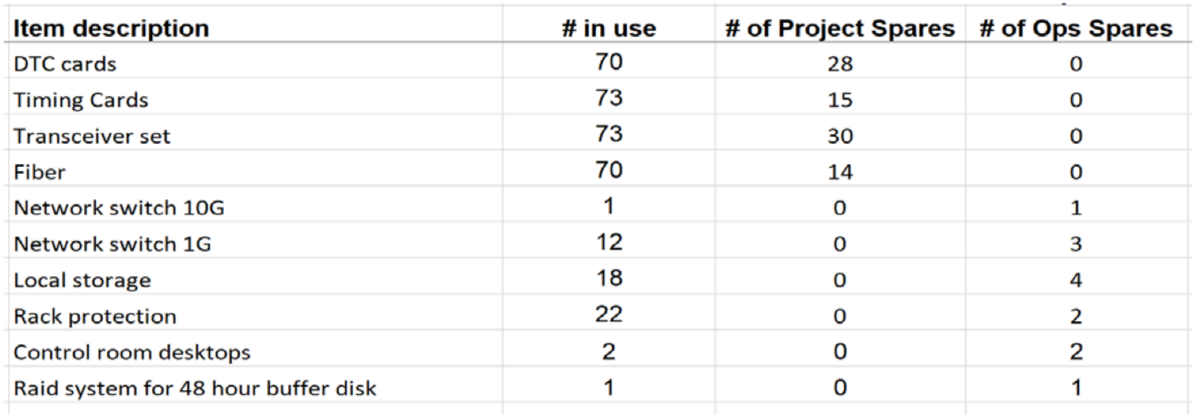


We have 69 DTCs and 1 CFO to host in 40 servers. The ExtMon/STM servers might have a single DTC → at most 3-4 servers will be DTC/CFO free.

The DTC/CFO free servers can fail without impacting the data integrity. Any other failure means we either lose data integrity, slow control monitoring or data taking capabilities. Not having enough spares is the risk.

Hardware risk
Mitigation strategy

Spare parts at the beginning of operation (project buy back + operation acquisition).



Item description	# in use	# of Project Spares	# of Ops Spares
DTC cards	70	28	0
Timing Cards	73	15	0
Transceiver set	73	30	0
Fiber	70	14	0
Network switch 10G	1	0	1
Network switch 1G	12	0	3
Local storage	18	0	4
Rack protection	22	0	2
Control room desktops	2	0	2
Raid system for 48 hour buffer disk	1	0	1
Servers from test stand*	10	5	0
New server	0	0	2

Note 1: the test stand remains alive for new developments / tests but some components are taken out when needed. We don't expect much development at a later stage.

Note 2: We plan to replace the servers during the long shutdown, so we should have enough spare servers for Run 1.

*Servers from test stand will already be quite old, and we want to have a few new ones just in case.

Hardware risk
Mitigation strategy (2)

Risk mitigation strategy

- Project already bought some spares and we believe this should be enough for the duration of operations (other expert opinion welcome) -> risk free

- For the remaining items, operations will only purchase a small number of spares and consider failures as risk.

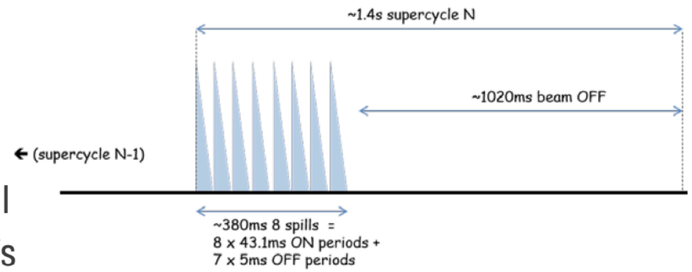
Event description	Technical Impact	Cost Impact (k$)
Failure of 10Gb network switch	Can't take data	30
Failure of 1Gb network switch	Part of TDAQ system inoperative	1
Failure of 48 hour buffer disk	Can't transfer data to tape	1
Failure of control room server	Can't pilot TDAQ from control room	1

Need to check with Ryan if I need to include something for local storage and rack protection.

Performance risk

Trigger performance

- ON Spill event contribution:
  - 43.1ms / 1695ns = 25K pulses per spill
  - 25K * 8 / 1.4s = 145K ON Spill events/s



- OFF Spill event contribution:
  - 1s / 30 us = 33K events.
  - Off-spill trigger very fast (if needed), so almost for free

The trigger algorithms should run in (assuming negligible off-spill contribution):

$$145\text{K}\ \frac{Events}{s},\ 40\ \text{nodes},\ 20\ \frac{art\ Threads}{Nodes} = 5.5\ \frac{ms}{\frac{Events}{art\ Threads}}$$

for 20 threads running simultaneously.

The current performance is 3.4ms on a fully threaded server for the track trigger.

We have versions of the trigger code that is slightly less efficient but much faster.

Performance risk and mitigation

Assuming we have optimized the code, we have two (cascading) risks:

1. The trigger performance is below the requirements (<90% efficiency)

   The mitigation is to install more servers - a good fraction of the TDAQ racks is empty, and we can increase the trigger farm by adding 40 more servers or so.

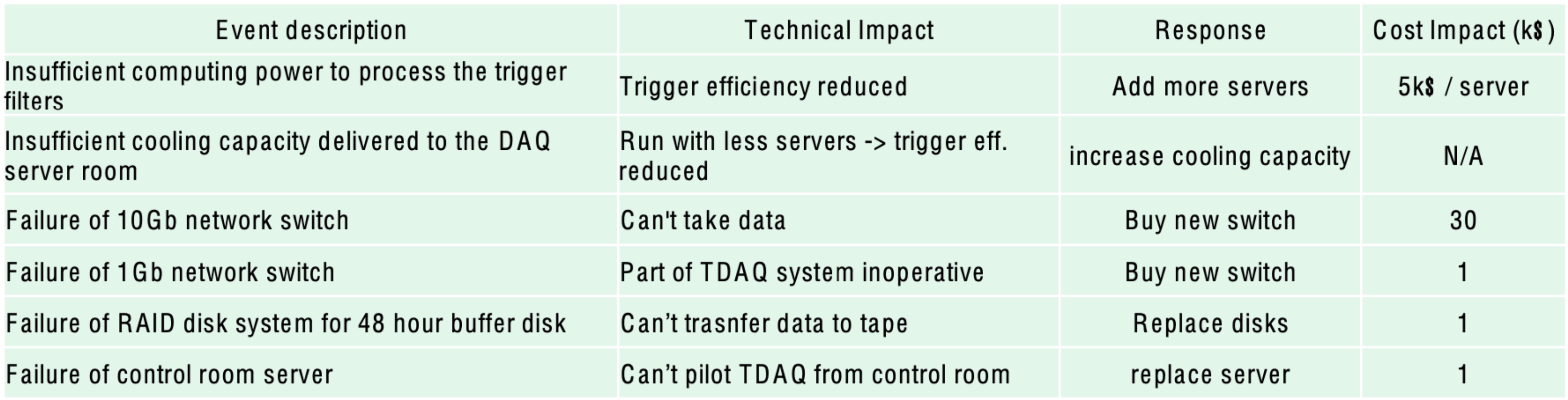
2. The TDAQ cooling power is insufficient

   We are currently using 83% of the total cooling capacity, and adding too many servers will require the installation of additional cooling power – likely new equipment in the TDAQ room.

Event description	Technical Impact	Cost Impact (k$)
Insufficient computing power to process the trigger filters	Trigger efficiency reduced	5k$ / server
Insufficient cooling capacity delivered to the DAQ server room	Run with less servers -> trigger eff. reduced	N/A

Full risk table

The current risk table:



Event description	Technical Impact	Response	Cost Impact (k$)
Insufficient computing power to process the trigger filters	Trigger efficiency reduced	Add more servers	5k$ / server
Insufficient cooling capacity delivered to the DAQ server room	Run with less servers -> trigger eff. reduced	increase cooling capacity	N/A
Failure of 10Gb network switch	Can't take data	Buy new switch	30
Failure of 1Gb network switch	Part of TDAQ system inoperative	Buy new switch	1
Failure of RAID disk system for 48 hour buffer disk	Can’t trasnfer data to tape	Replace disks	1
Failure of control room server	Can’t pilot TDAQ from control room	replace server	1

Will review the risk analysis over the next weeks and update the table if needed.

FastCalo module update Motivation

Lot of recent work on the DAQ of of things: lots of changes to the ArtFragment code in the artdaq old work:

- The ROC now produces additional information as a result layout of the artdaq::Fragments has changed
- The trigger algorithms can exploit this new information and run faster
- Now need to make the TDAQ-CaloFront Offline code for the quick reconstruction compatible with these changes.

So that some changes have been proposed for the FastCalo module which currently resides in CaloCluster of Offline.

- FastCalo Currently takes in the CaloDigi and produces CaloClusters, all in one module. Its aim is to provide quick reco for online purposes.
- Fast Calo reads digis which have been built from art fragments hits and clusters without applying all the Offline algorithms for reconstructing the hits. We want to keep it that way. Its overall functionality will not be changed.
- Fast Calo is used for trigger studies. Calo-Clustering is an important step in cal-pat-rec.
- Only hits making a cluster are actually stored! This could be bad for the DQM?
- We have not yet had chance to assess the algorithm before in a realistic environment. Offline is essentially a “placeholder” which we can now build on.

New Data products Changes

- In the online workflow, data fragments containing waveform peak information, error flags and event mode flags have been created to deal with the additional input data from the ROCs.

- There will need to be a change if data products:
  - NewCaloDigi
    - Contains: _roId(-1), _t0(0.), _waveform(0), peakpos(0.), _errorFlag(0), _eventMode(0) u Error flags: for data quality
    - Peak position: for fast estimate of total charge and hit time
    - Event mode: helps select calibration
  - NewCaloRecoDigi
    - Contains: Now contains art::Ptr<NewCaloDigi>

- As CaloCrystalHit, and CaloCluster depend on these they too are currently new versions....it might be do-able to use the old versions with some fine-tuning.

New Modules
Proposed changes

- Needs to occur in 3 stages → these still need to be concise and we don't need anything too complicated
  - Motivation: Inspecialruns, like the source calibration or the laser run, we want to do the hit reco to check that all the channels are working, but we will not need the clustering.
- The transitions:
- NewCaloDigiCollection → NewCaloRecoDigi
- NewCaloRecoDigi → NewCaloCrystalHit
- NewCaloCrystalHit → CaloCluster

So we need at least 2 new modules,at the moment 3 have been build :

- RecoDigiMakerModule:
- HitMakerModule:
- ClusteringModule:

New features usage

- Error flag:
  - Can skip a digi, fill space with empty digi?
  - We need to discuss with all stakeholders once we have this functionality
  - Need to decide what to do: do we want to just ignore it? Reconstruct a "dummy" reco hit? Depends on how wewant to implement the DQM for counting these corrupted hits...
- Event Mode:
  - Can use for specific calibration, for example, might want to use different reco algorithm for example in laser-calibration runs
  - In these runs, the shape of the signals will be different
  - We again need to discuss the use of this feature once the code is functioning to see how people might want to use it.
- Peak Position:
  - Avoid need for waveformprocessor -> this was not used in the previous version as the code had its own way of extracting the recodigi.
  - The ROC will put only the position of the highest peak here. We assume for now no pile-up.
  - Will be used in online reco for reconstructing the time and charge in the fastest way

How will this fit in with Calibration?

- We need to integrate two databases in these modules:
  - Calibration database (for converting the peak amplitude to energy and to correct the various time offsets)
  - Dead channels map
- These are extremely light databases copied locally on the DAQ servers, but we still need to discuss the bookkeeping