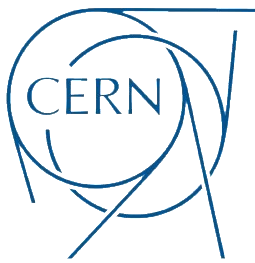


# **FESA & CMW - the outlook for thread-safe and real-time behaviour**

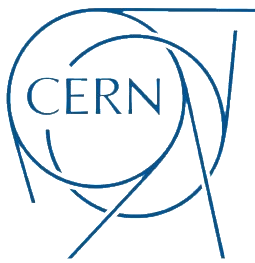
S. Deghaye BE/CO

Thanks to R. Steinhagen, M. Andersen, J. Wenninger, M. Lamont, NMN & W. Sliwinski for  
their inputs



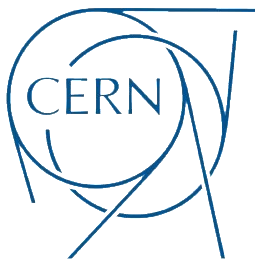
# Agenda

- Quick recap of the CO-related issues
- Technical details (FESA & CMW)
- Possible solutions
- OFSU special case



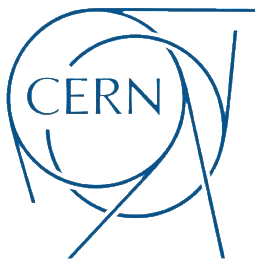
# CO-related issues

- Timing distribution of the energy
  - Threading in FESA 2.10
  - Greedy or bad CMW clients
  - VME Access
  - RT programming & Tools
- Thread-safe
- Real-time behaviour



# Timing distribution of the energy

- Energy info wrong at OFC level
- Probably a threading issue in TGM (multi-core CPU)
- Solved by getting the energy from the BPM crates
- TGM is obsolete → won't be fixed
- The new lib (TimDT) takes that into account



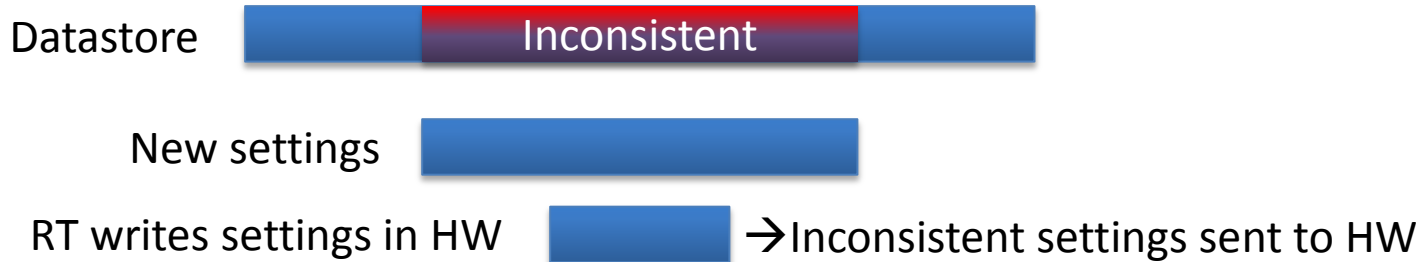
# Threading in FESA 2.10

- Visible in OFSU where settings were overwritten
- Known limitation of FESA 2.10 where server part and RT part step on each other
- Classic approach based on RT priorities is insufficient in a multi-core environment



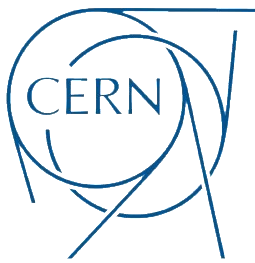
# Threading in FESA 2.10

- Settings case



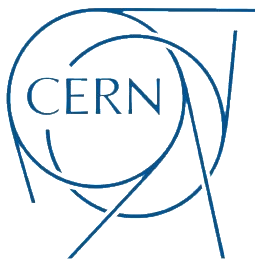
- Acquisition case





# Greedy or bad CMW clients

- Visible BPM front-ends (LynxOS)
- Heavy CMW load prevent RT execution causing  $\Delta$  latency on BPM samples
- Known problems in LynxOS:
  - priority inversion by boosting the com thread to the highest prio
  - Limited resources - mbuf
- Could be coupled with network issues too
  - see Ralph's QoS request to IT



# Greedy or bad CMW clients

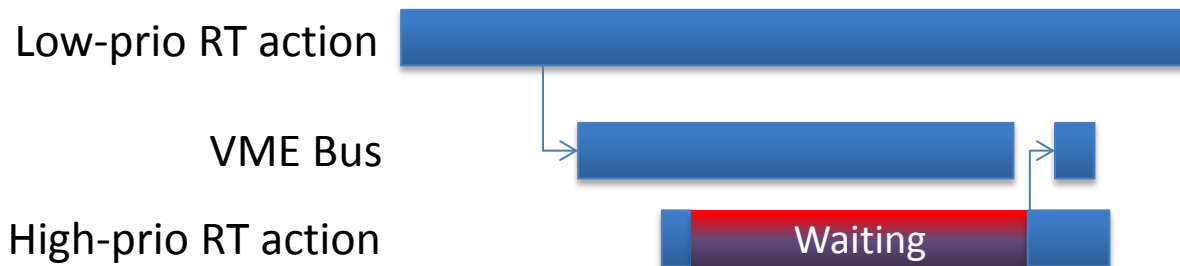
- Bad/slow clients can degrade the system in a similar way
- More difficult to take into account
- Proxy & RBAC rules put in place to shield the FECs

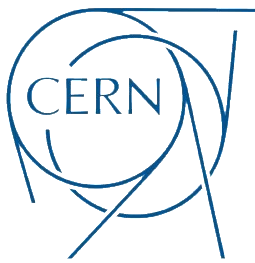




# RT priority & VME bus

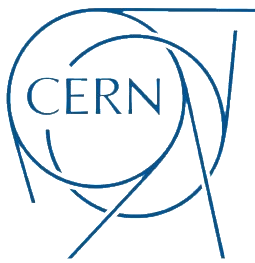
- Visible in the BPM crates
- Retrieval of large data blocks orbit data
- Priority inversion but at VME level





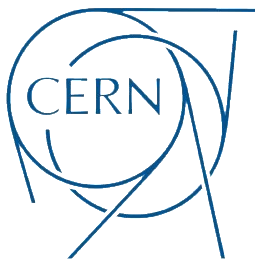
# Real Time Programming

- RT problems are difficult to understand
- FESA provides a profiling tool but it affects too much the normal load
- FESA lacks of
  - light-weight RT diagnostics
  - RT constraint specification and verification



# Possible solutions

- Threading in FESA3
- Greedy or bad CMW clients – new OS & CMW
- VME bus – new CPU
- FESA RT features – some proposals

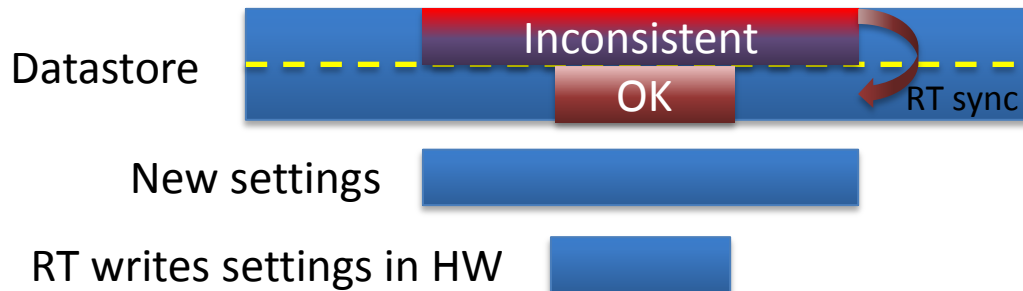


# Threading in FESA3

- FESA 2.10 Settings case



- FESA 3 double-buffer solution



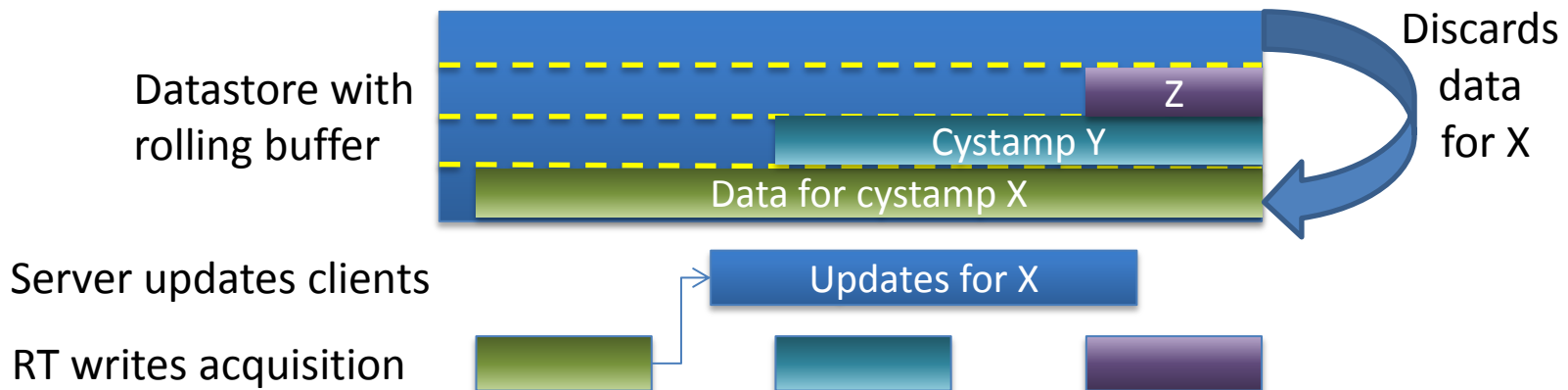


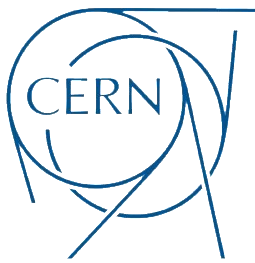
# Threading in FESA3

- FESA 2.10 Acquisitions case



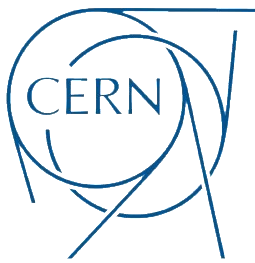
- FESA 3 rolling-buffer solution





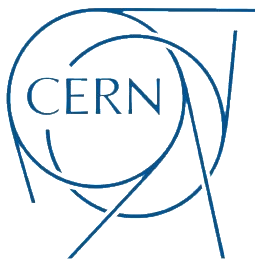
# Greedy or bad CMW clients

- Migration to MEN A20 foreseen → Linux as OS
- Solves the prio-boost problem but...
- Mind buffer usage... if TCP buffer full → thread waits
- Also, 2 Ethernet interfaces don't solve everything...
  - TCP/IP stack is common resource → same buffer issue
- RDA3:
  - Better scalability (prelim results)
  - Client priority support for all operations



# RT priority & VME bus

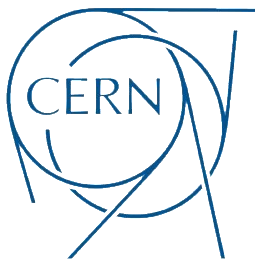
- This problem must be taken into account when writing the access code
- The good news:
  - MEN A20 has 2 VME channels (software)
  - Could allow interleaved access e.g. between 512-word block transfers (BLT DMA)
- But...
  - Requires more tests on CO side to really understand the possibilities



# Real Time in FESA

- Ralph: “can’t improve what I can’t measure”
- Possible FESA3 improvements:
  - Provide RT constraint specifications & lightweight RT constraint checks
  - Provide CPU Affinity settings
    - !! for FESA threads only → solve half of the problem
    - + Need to be sure of the effect → A. Rubini’s study

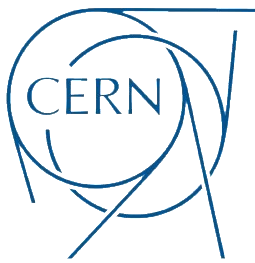




# Real Time in FESA

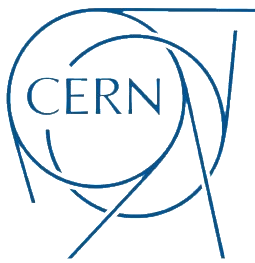
- Valgrind:
  - FESA pre-configured exceptions
  - e.g. 

```
{  
  String_S_create  
  Memcheck:Leak  
  fun:_Znwm  
  fun:_ZNSs4_Rep9_S_createEmmRKSaIcE  
}
```
  - work on proper deallocation on exit
- INTEL VTune amplifier & inspector
  - To be tried...



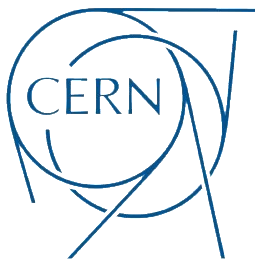
# OFSU special case

- Reported/seen as the weakest link
- FESA 2.10 based
- Big device data model
- Many properties + some are quite big (1MB+)
- Reasonable update rate (1 Hz)
- Big setting properties (3MB+)



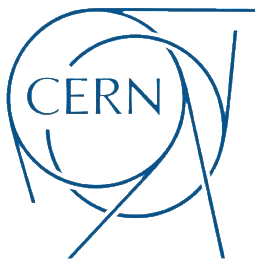
# FESA for OFSU?

- From the main developer: Yes
- But improvements in order
  - Flat data structure (FESA3 OK)
  - Threading issue (FESA3 + redesign settings and acquired settings)
- Update rate easily handled
- Amount of data OK
- Last two points to be reviewed as requirements are modified



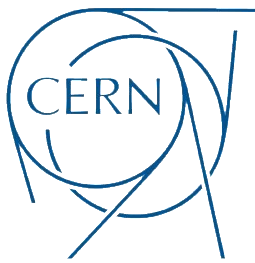
# OFSU – worth checking...

- Awkward configuration fetching
  - Start JAVA process
  - JAVA process uses LSA API & write in a file
  - The file is read by OFSU
  - OFSU sends the data to the OFC
- What about data corruption? CRC...
- Bi-directional settings (SRV to RT & RT to SRV)



# Things to do (I/II)

- MEN A20
  - Network prio boost in LynxOS
  - VME DMA (to be tested)
- FESA 3
  - Threading
  - Finer control of RT prio
  - RDA3 with better scalability & client priority
  - Other developer goodies (structures, ...)



# Things to do (II/II)

- OFSU
  - Configuration fetching (file transfer)
  - Settings flow: foresee a redesign