SVT DAQ Setup @ SLAC

PF

06/20/2021

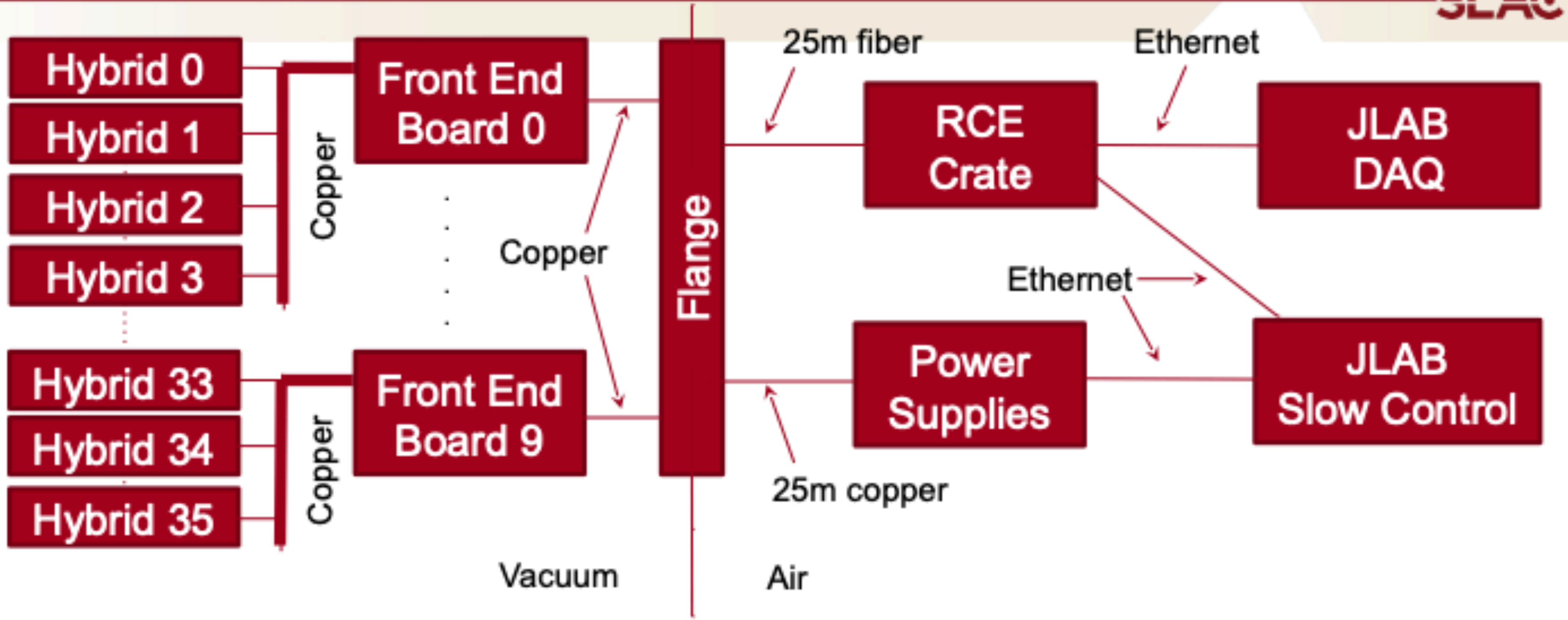
U.S. DEPARTMENT OF ENERGY | Stanford University

SLAC NATIONAL ACCELERATOR LABORATORY

Outline

- **Introduction**
  - SVT DAQ Overview
  - Some inspiration taken from Cam's Collab 2019 Talk
- **DAQ Setup at SLAC**
  - Architecture schematics
  - Hardware components
  - Back-end setup: current status
- **Local Data taking**
  - Feb testing
- **Integration in jLab system**
  - Integration with CODA and Epics
  - Slow Control integration / Monitoring integration
- **Next steps and plans**

SVT DAQ Overview



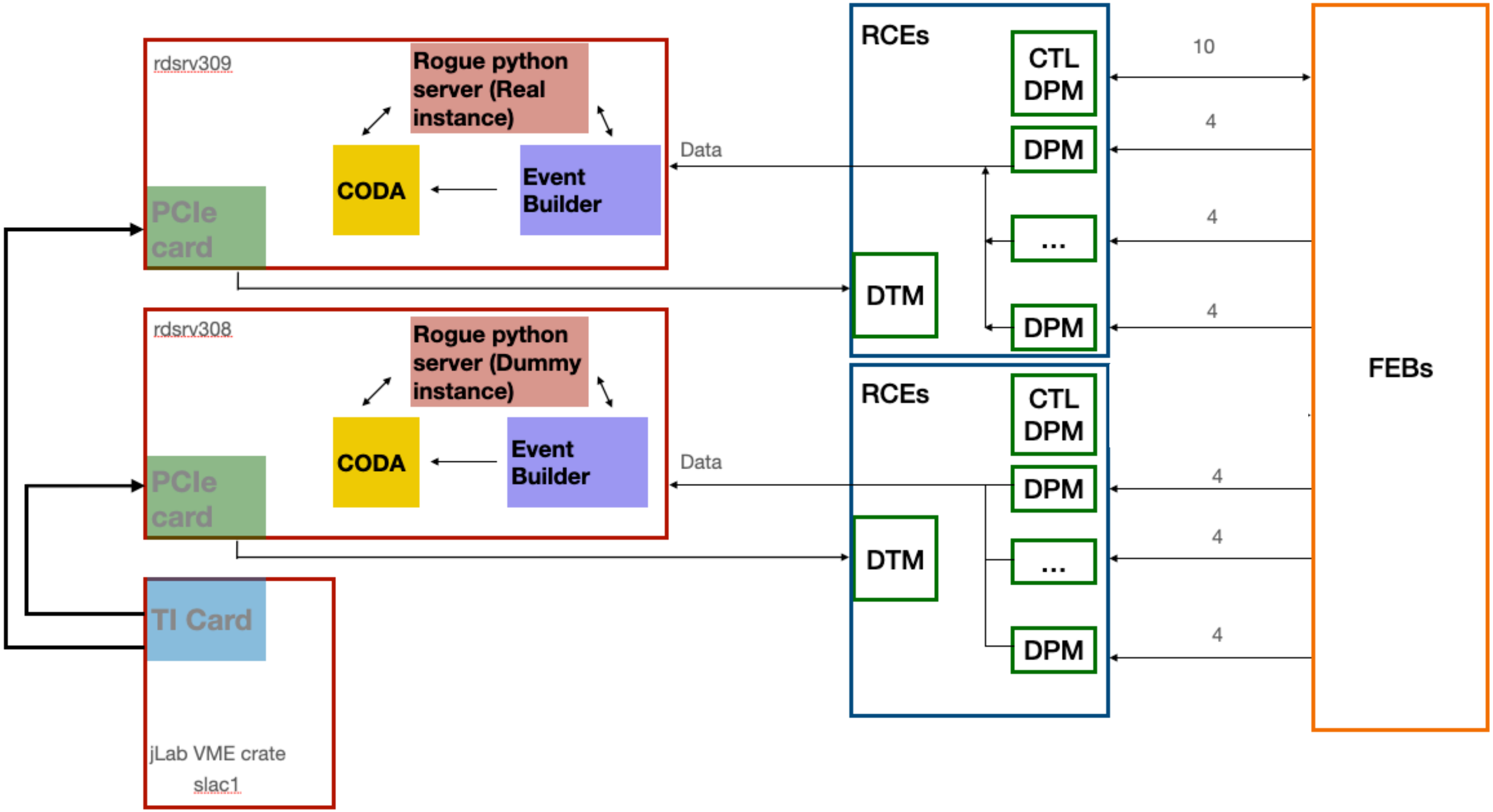
- 36 hybrids
  - 12 in layers 0 – 3 (2 per module)
  - 24 in layers 4 – 6 (4 per module)
- 10 front end boards
  - 4 servicing layers 0 – 3 with 4 hybrids per board
  - 6 servicing layers 4 – 6 with 4 hybrids per board
- RCE crate: ATCA, data reduction, event building and JLab DAQ interface

Raw ADC data rate (Gbps)	
Per hybrid	3.33
Per L1-3 Front end board	10
Per L4-6 Front end board	13

SVT RCE Allocation - 2019 schematics

- Two COBs utilised in the SVT readout system
  - Total 16 RCEs (Data and Control DPMs, 8 DPM per COB)
  - 2 DTM (1 DTM per COB)
- 7 RCEs on each COB process data from half SVT
- 8th RCE on COB 0 manages all 10 FEBs Board
  - Configuration / Slow controls / Status monitoring
  - Clock and trigger distribution to FE boards / Hybrids
- 8th RCE on COB 1 is not used

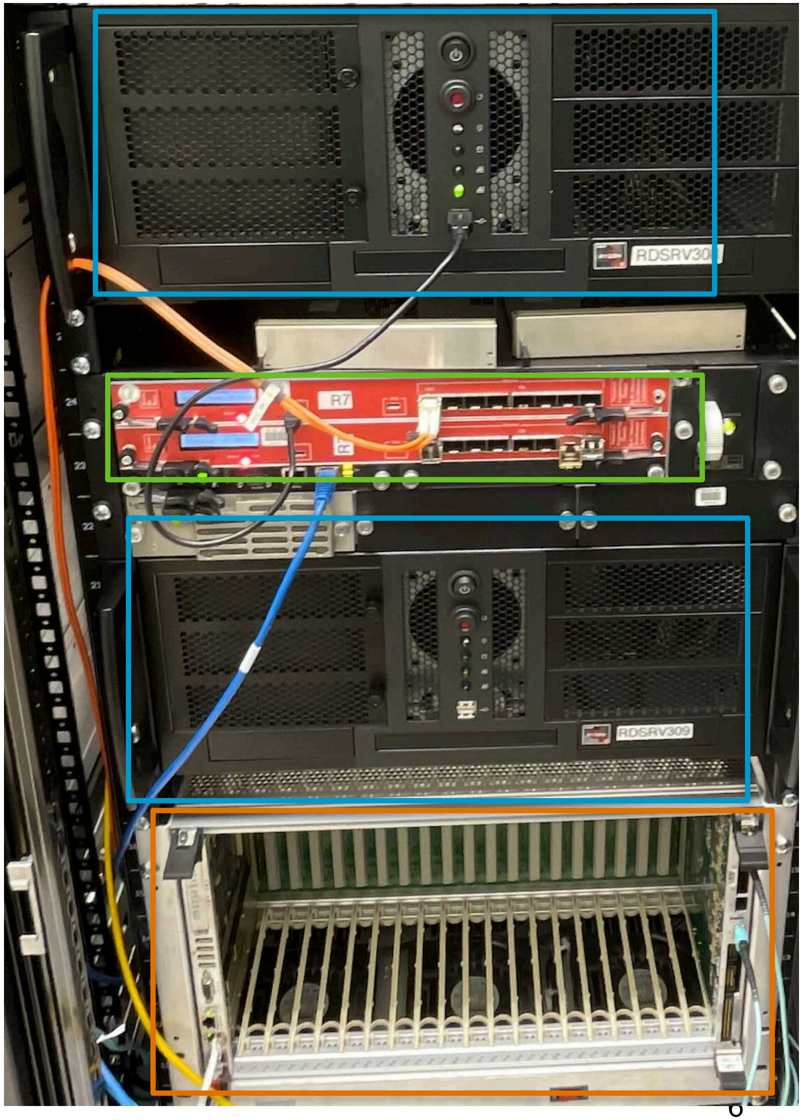
DAQ Infrastructure architecture



- Schematics of the SVT DAQ Architecture

Hardware components - ATCA crate - Front

- 2 server machines rdsrv309/308 connected
- Internal daq network between server machines, VME TI Crate and RCEs working and fully functional
  - Network sharing in the backplane to be cut
- The 2TiPCIe cards have been installed
  - Fiber connection from TI crate
  - Ribbon cable connection to RCEs
- Slow controls to the test stand in SLAC



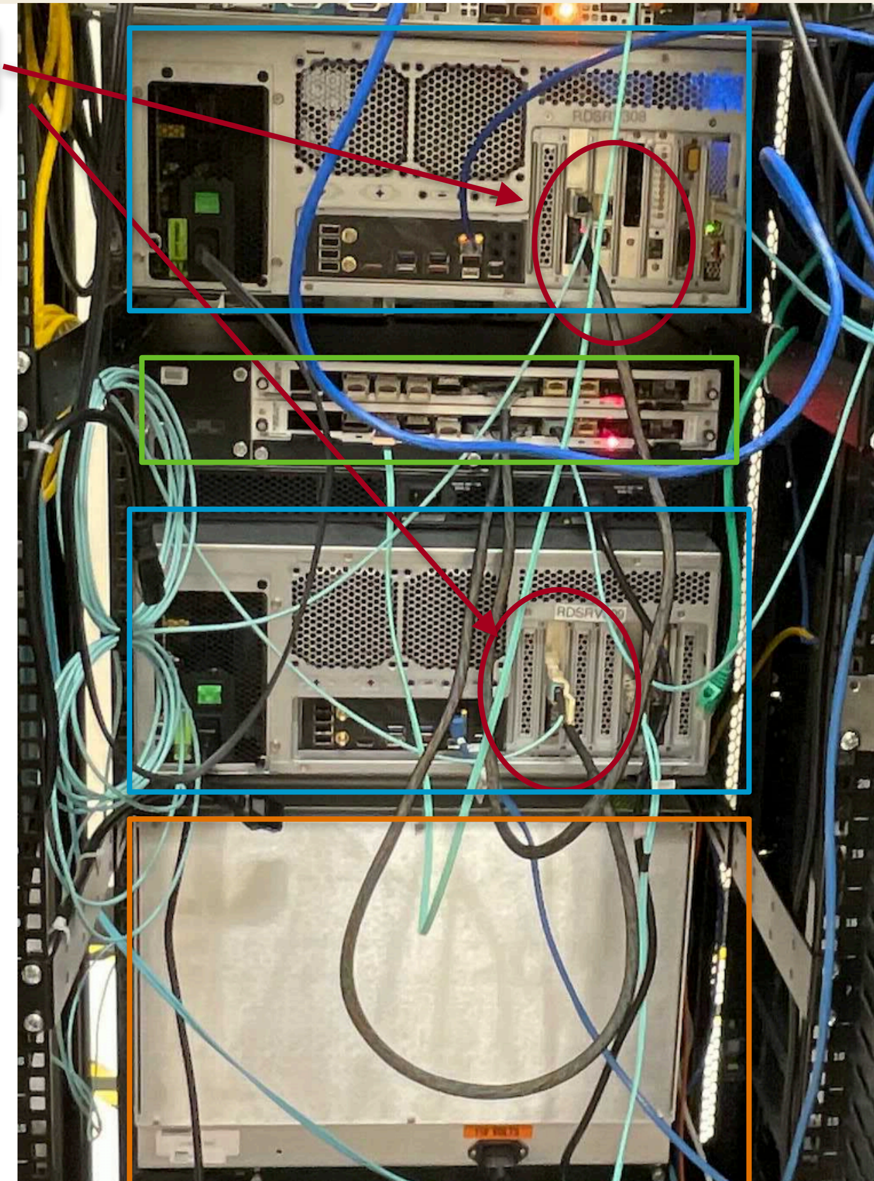
Servers 309/308

RCEs
Bottom COB:
slow ctrl / Data

VME TI Crate

Hardware components - ATCA crate - Back

- 2 server machines rdsrv309/308 connected
- Internal daq network between server machines, VME TI Crate and RCEs working and fully functional
  - Network sharing in the backplane to be cut
- The 2TiPCIe cards have been installed
  - Fiber connection from TI crate
  - Ribbon cable connection to RCEs
- Slow controls to the test stand in SLAC



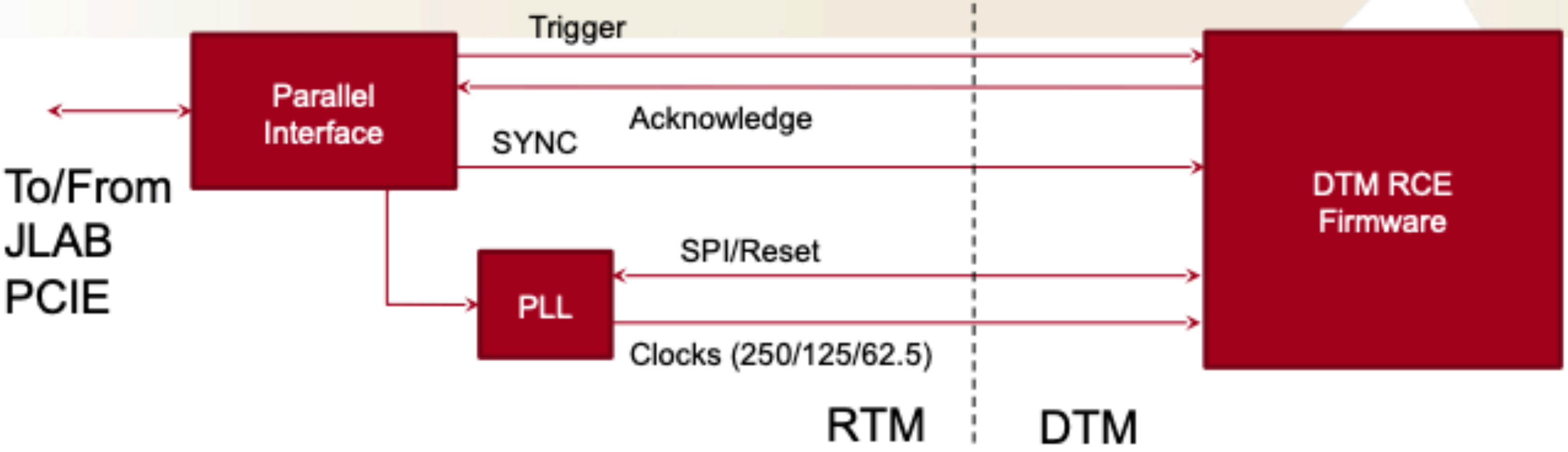
TiPCIe Cards

Servers 309/308

RCEs
Bottom COB:
slow ctrl / Data

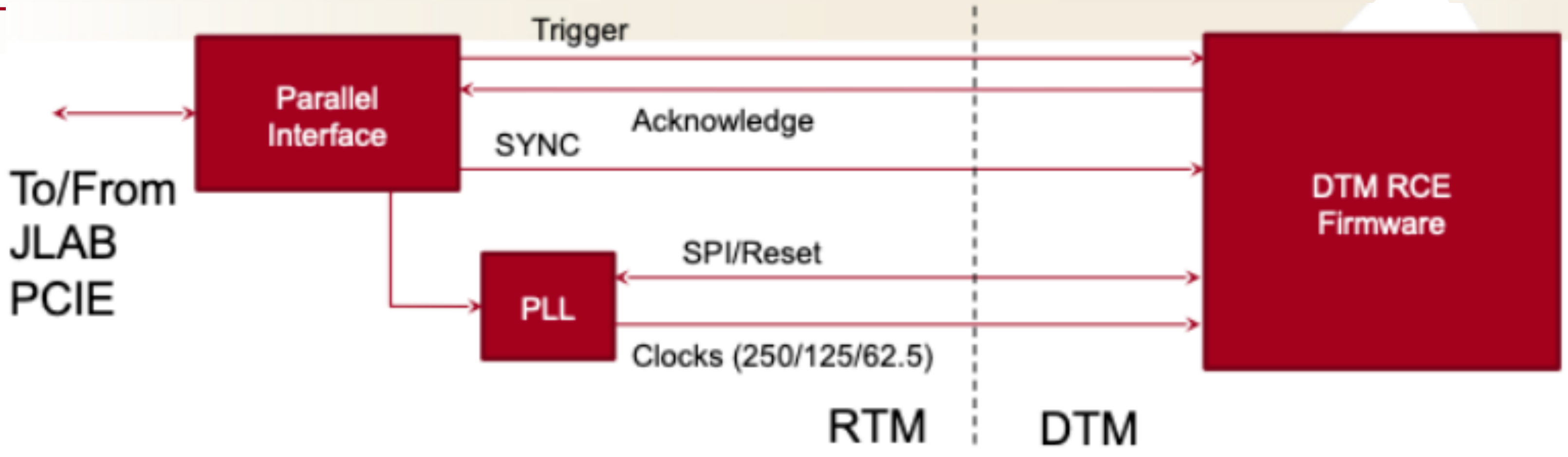
VME TI Crate

SVT Trigger Interface



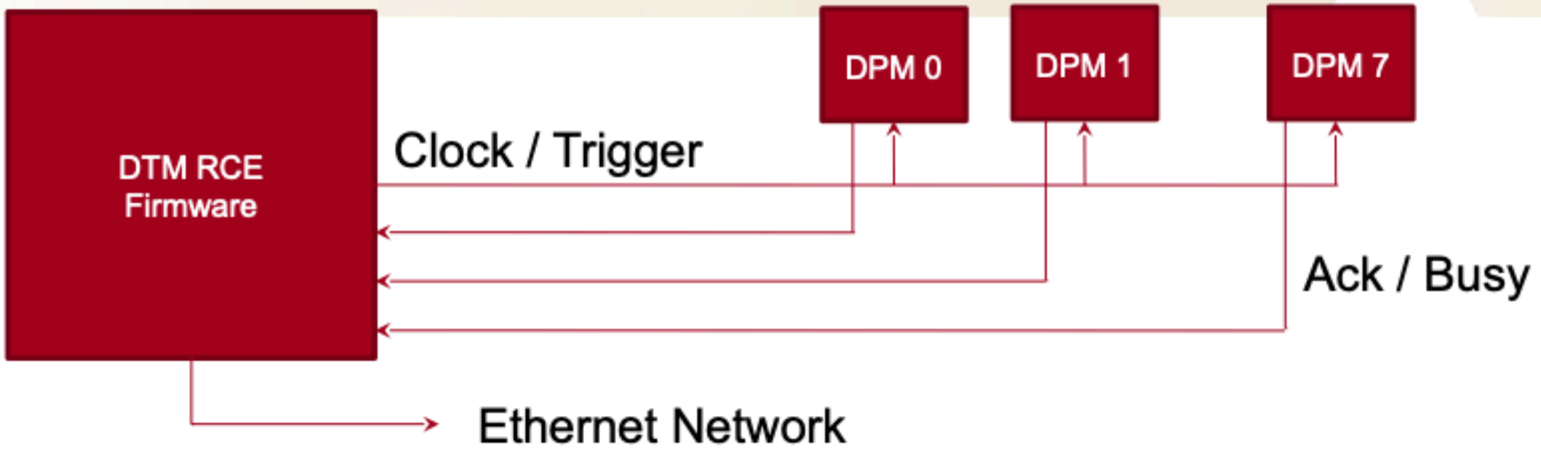
- Parallel interface and PLL exist on new RTM
- Fully allocated available signals between RTM and DTM
  - 1 high speed pair for trigger & SYNC
  - 1 low speed pair for SYNC
  - 2 low speed pairs for PLL SPI and Reset signals
  - 3 low speed pairs for PLL generated clocks (250/125/62.5 Mhz)

Current Firmware Developments



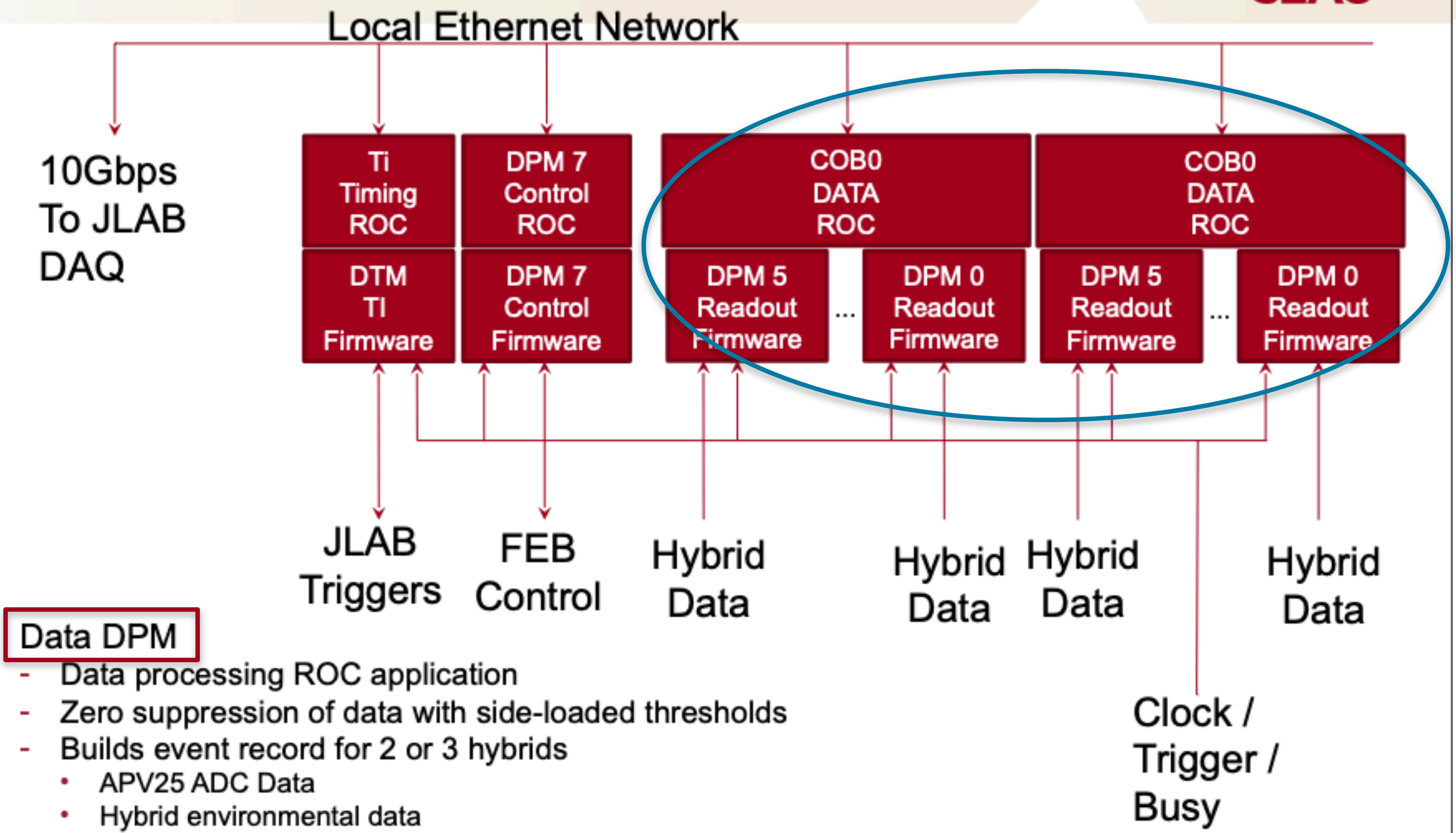
- Observed in the past that sync signal being sent to the DTMs from TIPCIe was out of phase
- This leads to an out of phase sync signal being distributed to the other RCE components
- The full test stand setup at SLAC will be used to develop a FW to cancel out the phase difference in the sync signals

SVT Trigger Distribution



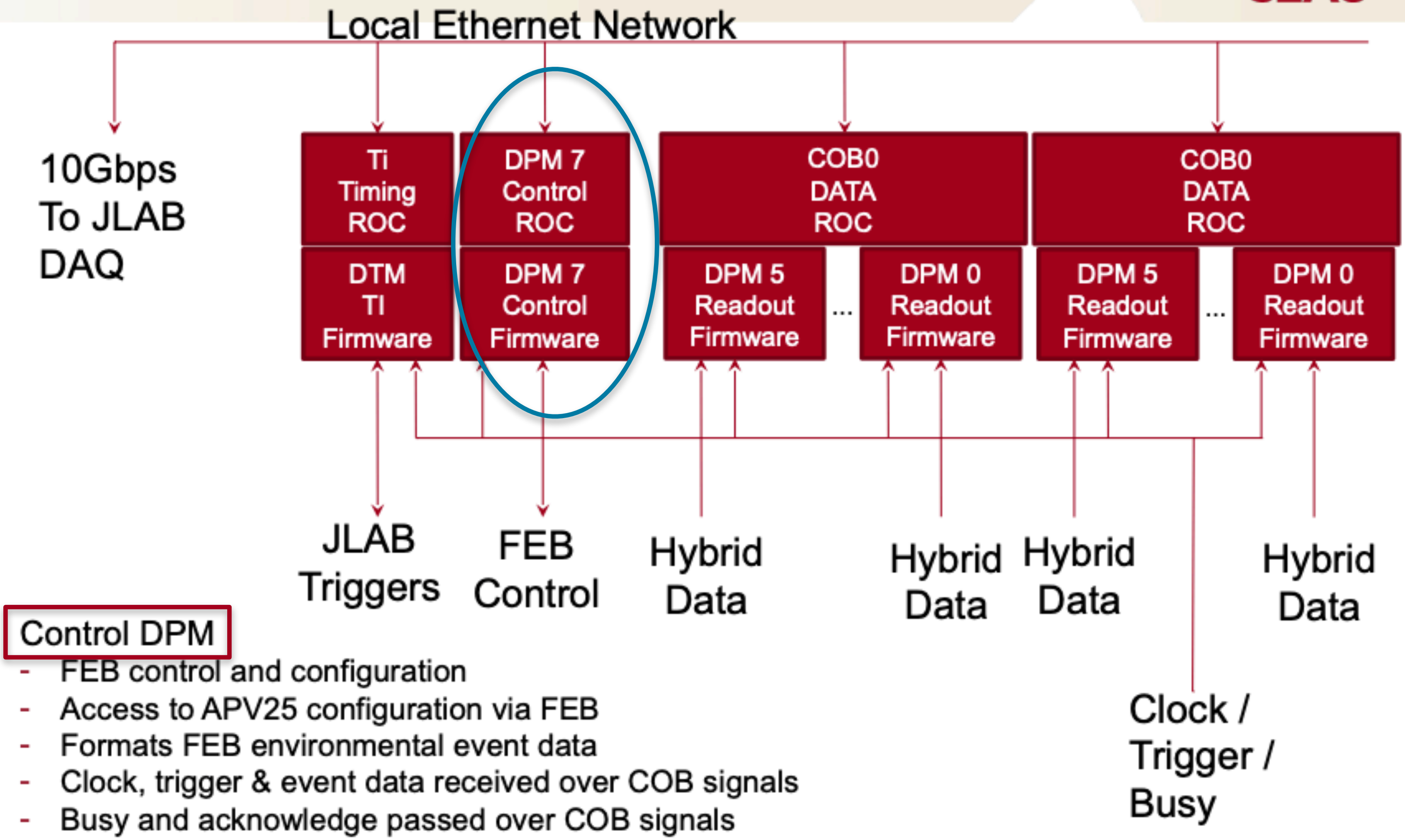
- DTM FPGA has ability to distribute clock and trigger to DPMs
  - Clock and trigger wired as fan out to DPMs
  - Individual feedback signals from each DPM
- 1 pair for clock fan out
- 1 pair for trigger fan out
  - 125Mhz serial protocol transfers 8-bit codes (easily expanded to longer words)
  - Used to distribute event codes to DPMs
    - System clock sync, APV25 sync & JLAB triggers
- 1 pair for trigger data distribution
  - Event and block data
- 1 pair per DPM for feedback
  - Readout and trigger acknowledge
  - Busy

ROC Instances On SVT



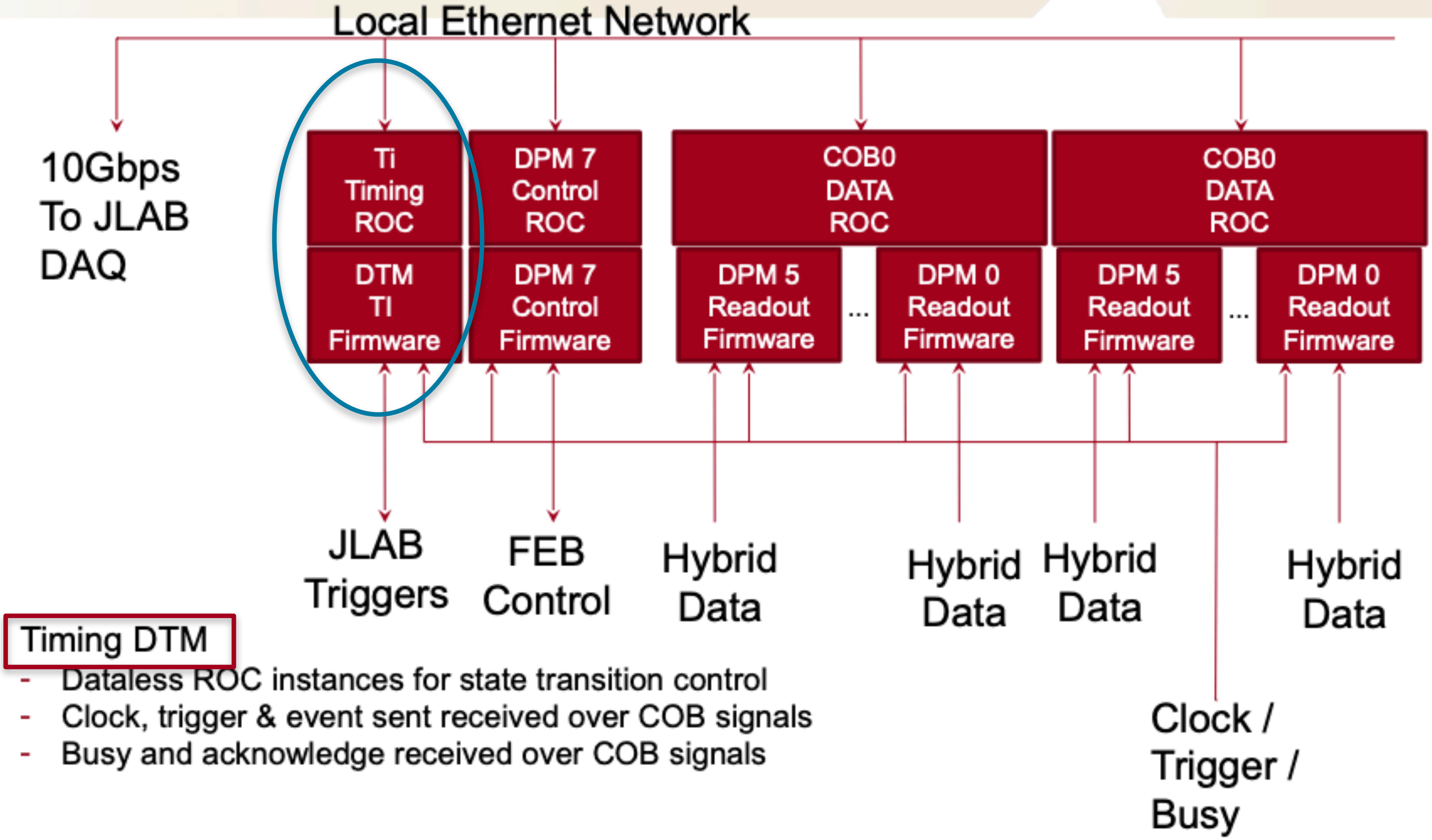
- Data DPM
  - Data processing ROC application
  - Zero suppression of data with side-loaded thresholds
  - Builds event record for 2 or 3 hybrids
    - APV25 ADC Data
    - Hybrid environmental data
  - Clock, trigger & event data received over COB signals
  - Busy and acknowledge passed over COB signals

ROC Instances On SVT



- Control DPM
  - FEB control and configuration
  - Access to APV25 configuration via FEB
  - Formats FEB environmental event data
  - Clock, trigger & event data received over COB signals
  - Busy and acknowledge passed over COB signals

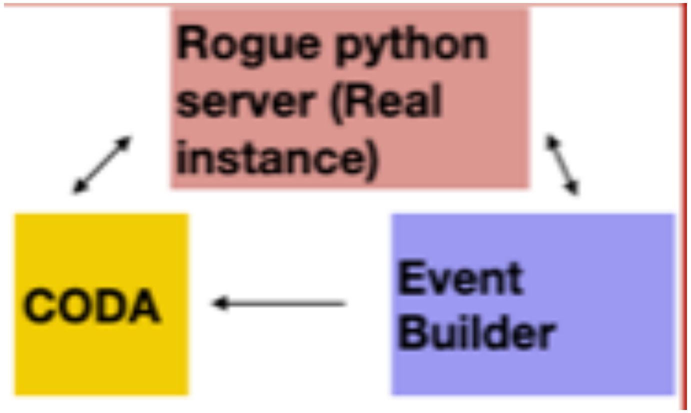
ROC Instances On SVT



- Timing DTM
  - Dataless ROC instances for state transition control
  - Clock, trigger & event sent received over COB signals
  - Busy and acknowledge received over COB signals

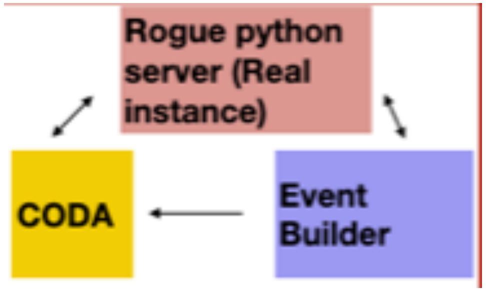
Back End - Software applications

- Rogue:
  - General DAQ interface for Configuration / Register monitoring of the whole COB+FEB system
    - Local data taking for quick diagnostics
    - Inter-process communications: CODA, Epics..
  - Updated to newest release from TID-AIR-ES (SLAC)
    - v5.8.0 (stable)
  - Minor changes for HPS Run 2021 interface



Back End Setup - Integration status

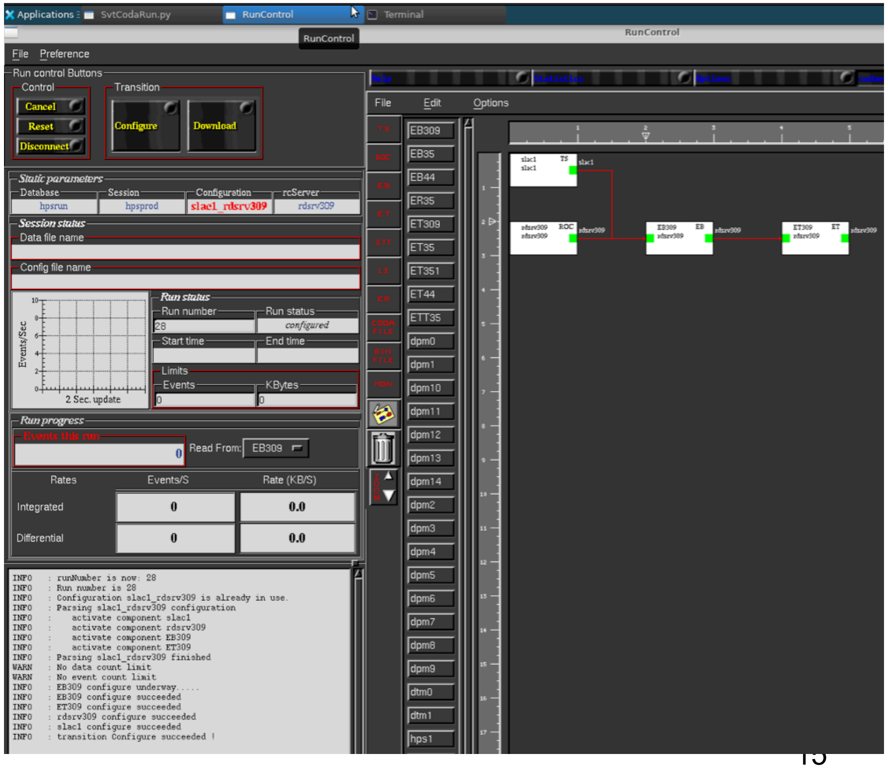
- All current hps daq sw collected here: [heavy-photon-daq GIT]
- Repository updated and compatible with newest version of Rogue v5.8.0
- Event-Builder compiled successfully against Rogue-Lite (i.e. rogue without python bindings) v5.8.0 (only minor modifications needed so far)
- Sergey successfully compiled and installed CODA in the server machines here at SLAC



Rogue python server (Real instance)

CODA

Event Builder



Static parameters

Database	Session	Configuration	rcServer
hpsrun	hpsprod	slac1_rdsrv309	rdsrv309

Session status

Data file name

Config file name

Run status

Run number	Run status
28	configured

Start time

End time

Limits

Events	KBytes
0	0

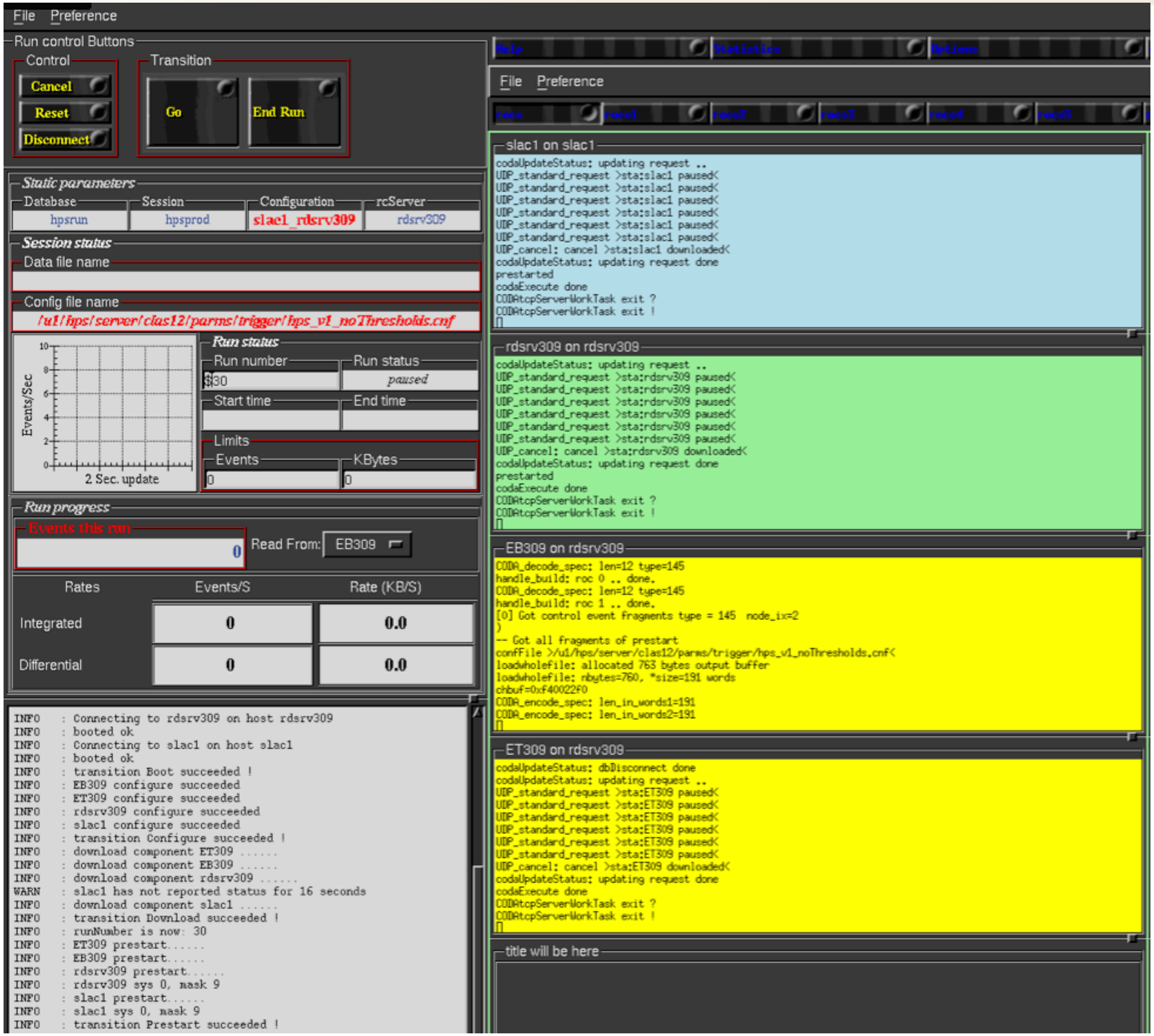
Events this run: 0

Read From: EB309

Rates	Events/S	Rate (KB/S)
Integrated	0	0.0
Differential	0	0.0

```
INFO   : runNumber is now: 28
INFO   : Run number is 28
INFO   : Configuration slac1_rdsrv309 is already in use.
INFO   : Parsing slac1_rdsrv309 configuration
INFO   :    activate component slac1
INFO   :    activate component rdsrv309
INFO   :    activate component EB309
INFO   :    activate component ET309
INFO   : Parsing slac1_rdsrv309 finished
WARN   : No data count limit
WARN   : No event count limit
INFO   : EB309 configure underway.....
INFO   : EB309 configure succeeded
INFO   : ET309 configure succeeded
INFO   : rdsrv309 configure succeeded
INFO   : slac1 configure succeeded
INFO   : transition Configure succeeded !
```

Back End Setup - Integration status - CODA



Static parameters

Database	Session	Configuration	rcServer
hpsrun	hpsprod	slac1_rdsrv309	rdsrv309

Session status

Data file name:

Config file name: /u1/hps/server/clas12/parms/trigger/hps_v1_noThresholds.cnf

Run status

Run number: $30

Run status: *paused*

Start time:

End time:

Limits — Events: 0 — KBytes: 0

Run progress

Events this run: 0

Read From: EB309

Rates	Events/S	Rate (KB/S)
Integrated	0	0.0
Differential	0	0.0

```
INFO   : Connecting to rdsrv309 on host rdsrv309
INFO   : booted ok
INFO   : Connecting to slac1 on host slac1
INFO   : booted ok
INFO   : transition Boot succeeded !
INFO   : EB309 configure succeeded
INFO   : ET309 configure succeeded
INFO   : rdsrv309 configure succeeded
INFO   : slac1 configure succeeded
INFO   : transition Configure succeeded !
INFO   : download component ET309 ......
INFO   : download component EB309 ......
INFO   : download component rdsrv309 ......
WARN   : slac1 has not reported status for 16 seconds
INFO   : download component slac1 ......
INFO   : transition Download succeeded !
INFO   : runNumber is now: 30
INFO   : ET309 prestart......
INFO   : EB309 prestart......
INFO   : rdsrv309 prestart......
INFO   : rdsrv309 sys 0, mask 9
INFO   : slac1 prestart......
INFO   : slac1 sys 0, mask 9
INFO   : transition Prestart succeeded !
```

slac1 on slac1

```
codaUpdateStatus: updating request ..
UDP_standard_request >sta:slac1 paused<
UDP_standard_request >sta:slac1 paused<
UDP_standard_request >sta:slac1 paused<
UDP_standard_request >sta:slac1 paused<
UDP_standard_request >sta:slac1 paused<
UDP_standard_request >sta:slac1 paused<
UDP_cancel: cancel >sta:slac1 downloaded<
codaUpdateStatus: updating request done
prestarted
codaExecute done
CODAtcpServerWorkTask exit ?
CODAtcpServerWorkTask exit !
```

rdsrv309 on rdsrv309

```
codaUpdateStatus: updating request ..
UDP_standard_request >sta:rdsrv309 paused<
UDP_standard_request >sta:rdsrv309 paused<
UDP_standard_request >sta:rdsrv309 paused<
UDP_standard_request >sta:rdsrv309 paused<
UDP_standard_request >sta:rdsrv309 paused<
UDP_standard_request >sta:rdsrv309 paused<
UDP_cancel: cancel >sta:rdsrv309 downloaded<
codaUpdateStatus: updating request done
prestarted
codaExecute done
CODAtcpServerWorkTask exit ?
CODAtcpServerWorkTask exit !
```

EB309 on rdsrv309

```
CODA_decode_spec: len=12 type=145
handle_build: roc 0 .. done.
CODA_decode_spec: len=12 type=145
handle_build: roc 1 .. done.
[0] Got control event fragments type = 145  node_ix=2
)
-- Got all fragments of prestart
confFile >/u1/hps/server/clas12/parms/trigger/hps_v1_noThresholds.cnf<
loadwholefile: allocated 763 bytes output buffer
loadwholefile: nbytes=760, *size=191 words
chbuf=0xf40022f0
CODA_encode_spec: len_in_words1=191
CODA_encode_spec: len_in_words2=191
```

ET309 on rdsrv309

```
codaUpdateStatus: dbDisconnect done
codaUpdateStatus: updating request ..
UDP_standard_request >sta:ET309 paused<
UDP_standard_request >sta:ET309 paused<
UDP_standard_request >sta:ET309 paused<
UDP_standard_request >sta:ET309 paused<
UDP_standard_request >sta:ET309 paused<
UDP_standard_request >sta:ET309 paused<
UDP_cancel: cancel >sta:ET309 downloaded<
codaUpdateStatus: updating request done
codaExecute done
CODAtcpServerWorkTask exit ?
CODAtcpServerWorkTask exit !
```

- Integration in CODA almost completed
  - Basic configurations have been deployed for baseline data taking
  - Currently working to complete the transition to GO state (Prestart Working OK)
  - Currently:

    - No SaveState in GO / EndOfRun

    - No loading of thresholds in Prestart Phase (DAQ Map need to be fixed for SLAC test-stand)

    - Check of the Configuration is disabled due to issues in read-back of the FEB registers
- Some modifications to the rogue python part needed

Back End Setup - Integration status - EPICS

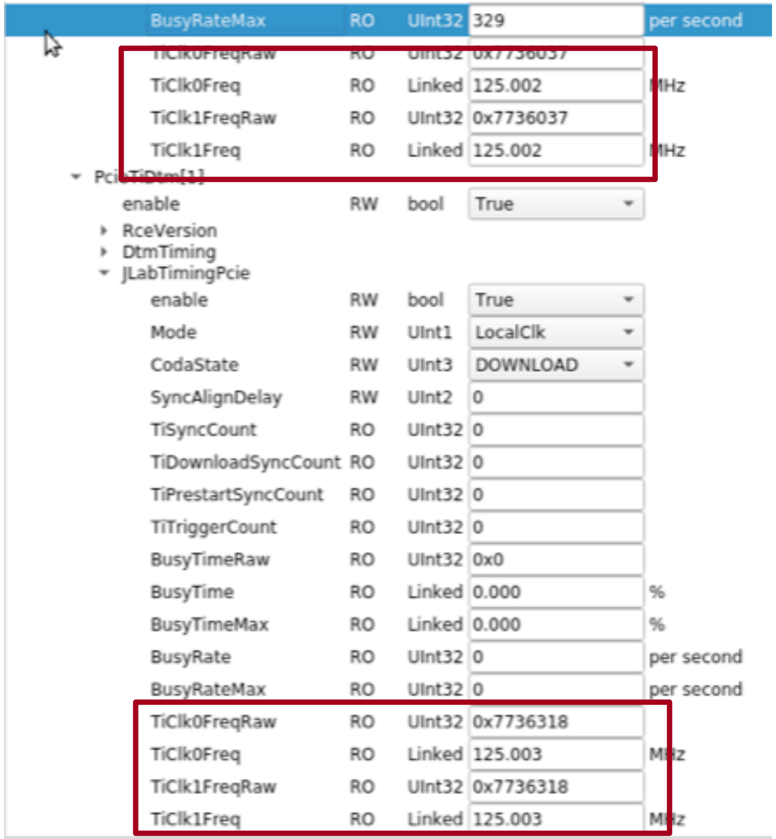
- Integration with EPICS
  - Only Basic checks performed
    - Verified communication between Rogue and EPICS:

      - I could obtain via “caget” variables pushed by Rogue with EPICS

      - Will prepare a full configuration for a IOC Machine to test that all is functioning as expected before the run

Current Issues - TO DO

- Some checks need to be performed on the clock from the TIPCIe card:
  - We are able to read the external clock frequency on the DTMs
  - We are able to configure and (partially) talk to the FEBs after configuring with external clock - we see the upTime OK -> pgp connection seems established
  - We receive corrupted data when performing read-back at pre-start stage:
    - Need to discuss with Ben/Ryan the source of the issue



BusyRateMax	RO	UInt32	329	per second
TiClk0FreqRaw	RO	UInt32	0x7736037	
TiClk0Freq	RO	Linked	125.002	MHz
TiClk1FreqRaw	RO	UInt32	0x7736037	
TiClk1Freq	RO	Linked	125.002	MHz
PcieTiDtm[1]				
enable	RW	bool	True	
RceVersion				
DtmTiming				
JLabTimingPcie				
enable	RW	bool	True	
Mode	RW	UInt1	LocalClk	
CodaState	RW	UInt3	DOWNLOAD	
SyncAlignDelay	RW	UInt2	0	
TiSyncCount	RO	UInt32	0	
TiDownloadSyncCount	RO	UInt32	0	
TiPrestartSyncCount	RO	UInt32	0	
TiTriggerCount	RO	UInt32	0	
BusyTimeRaw	RO	UInt32	0x0	
BusyTime	RO	Linked	0.000	%
BusyTimeMax	RO	Linked	0.000	%
BusyRate	RO	UInt32	0	per second
BusyRateMax	RO	UInt32	0	per second
TiClk0FreqRaw	RO	UInt32	0x7736318	
TiClk0Freq	RO	Linked	125.003	MHz
TiClk1FreqRaw	RO	UInt32	0x7736318	
TiClk1Freq	RO	Linked	125.003	MHz

Status Summary

- We are completing an SVT DAQ mock-up setup here at SLAC
- Several people contributed, supported and helped so far.
- Currently we are fixing some hiccups..

  - Some rogue hardcodes made for the jLab machines

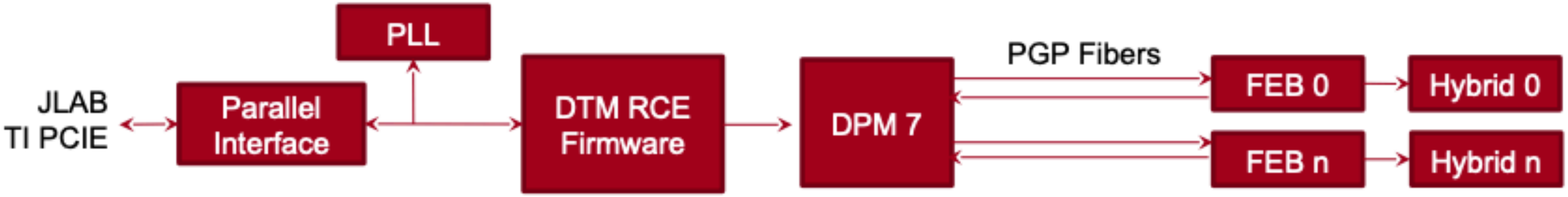
  - Daq Mapping should be configured properly for the small system we are testing here
- .. and some more complex issues:

  - Issues in reading back FEb configuration when using an external clock

  - Full transition to the GO state to be tested
- As soon as CODA is up and running (so the FW developers can work on the sync issue), I’m planning to switch to check EPICs integration

BACKUP

Front End Timing Distribution



- Control DPM forwards timing information to front end boards over PGP
  - Clock encoded into serial data stream which the front end board recovers
  - Fixed latency path for encoded PLL reset and trigger signals

BACKUP

BACKUP