# Discussion on processing time-frame data

Marco Battaglieri (JLab), Markus Diefenthaler (JLab), Jin Huang (BNL), Jeff Landgraf (BNL), Torre Wenaus (BNL)
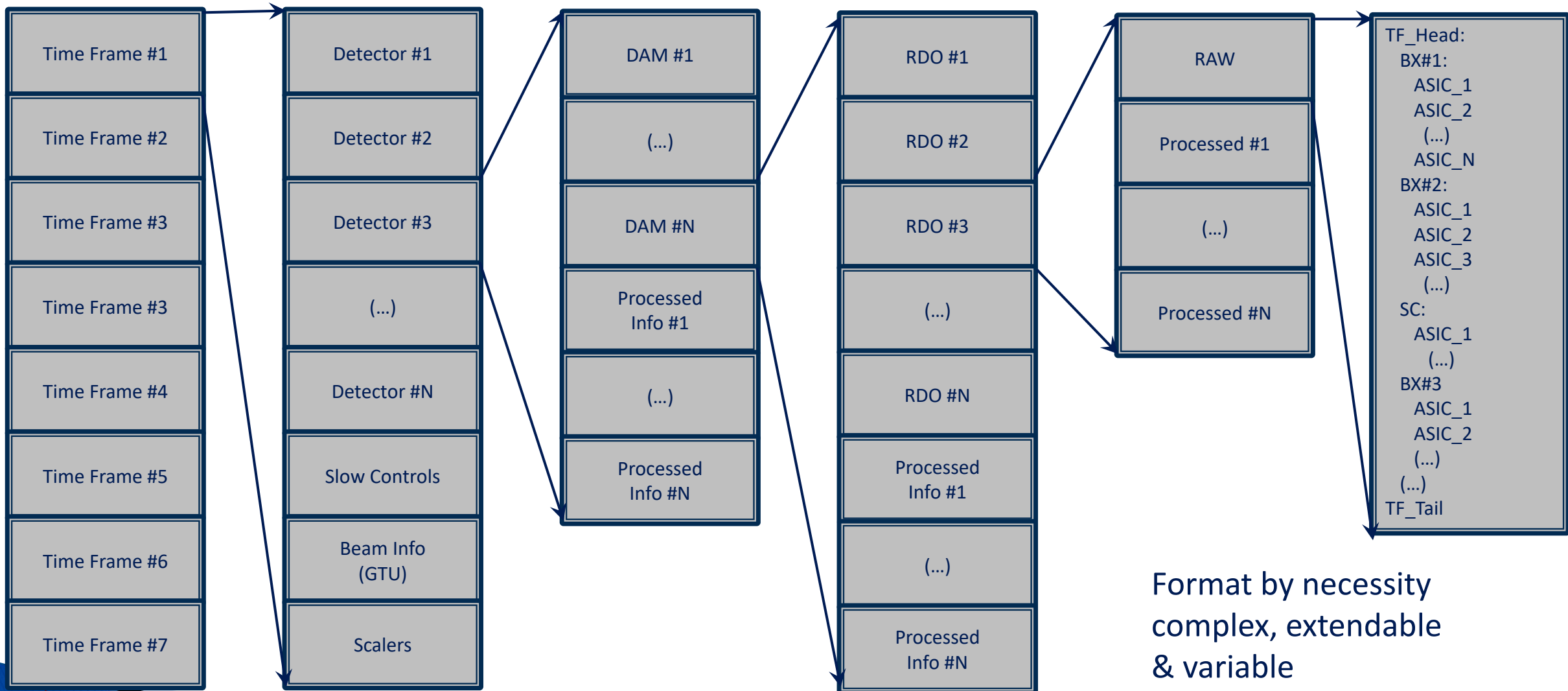
ePIC

Time-Frames Introduction

▸ We plan to use this meeting to follow up on Nathan's talk on time-frame-based reconstruction, solidify a few open concept in our WG and make progress on their implementation in EICRecon

▸ ePIC Time Frame concept is developing towards a spec doc in DAQ and SAR WGs;
  ◦ Update discussion on Apr 11 DAQ meeting , please join: https://indico.bnl.gov/event/22945/
  ◦ <=2^16 crossing: 16-bit integer sufficient to locate hit's BX in Time Frame; <=665us/300 events/10MB
  ◦ Exact length defined by GTU sync signal: most flexible
  ◦ We could choose to align with EIC beam evolution (1260BX,): simpler to locate abort gap and spin states

▸ Time Frames will be order in data files, internally carry header-payload (a.k.a data bank/packets) data chunks from each detector component.
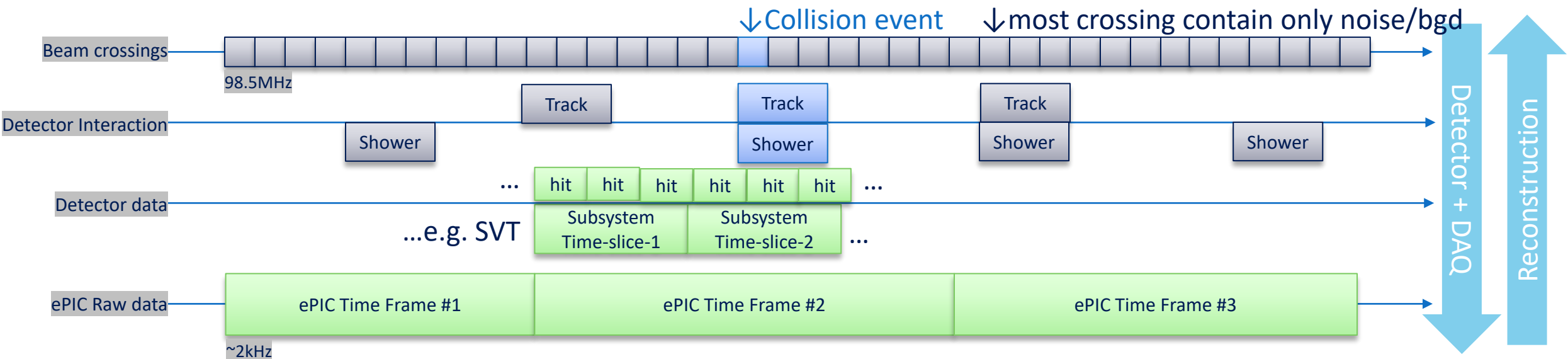
DAQ File Organization (Example...)

From Mar-21 meeting, Jeff's talk on Time Frame Organization and Data Volumes [link]

| Time Frame #1 |
| Time Frame #2 |
| Time Frame #3 |
| Time Frame #3 |
| Time Frame #4 |
| Time Frame #5 |
| Time Frame #6 |
| Time Frame #7 |

| Detector #1 |
| Detector #2 |
| Detector #3 |
| (...) |
| Detector #N |
| Slow Controls |
| Beam Info (GTU) |
| Scalers |

| DAM #1 |
| (...) |
| DAM #N |
| Processed Info #1 |
| (...) |
| Processed Info #N |

| RDO #1 |
| RDO #2 |
| RDO #3 |
| (...) |
| RDO #N |
| Processed Info #1 |
| (...) |
| Processed Info #N |

| RAW |
| Processed #1 |
| (...) |
| Processed #N |

TF_Head:
  BX#1:
    ASIC_1
    ASIC_2
    (...)
    ASIC_N
  BX#2:
    ASIC_1
    ASIC_2
    ASIC_3
    (...)
  SC:
    ASIC_1
    (...)
  BX#3
    ASIC_1
    ASIC_2
    (...)
  (...)
TF_Tail

Format by necessity complex, extendable & variable

Readers

Two distinct sets of readers needed

- Data Bank Navigation

  rdr = getBank("NameOfBank")   or
  rdr = getBank(TimeFrame, "lfhcal/dam_3/rdo_6/raw")

- Detector Bank specific readers (presumably implemented as plugins)

  hit = rdr->nextHit()
  hit.bx
  hit.highResTOA
  hit.channel
  hit.adc

- Could, of course have multiple readers instantiated at a time for simultaneous decoding
- One likely needs to fill intermediate data structure for processing, so time frame for DAQ and time frame for tracking need not be tied together!

Discussion 1: event keying

▸ One way to view information provided by streaming DAQ is clock triggered events at *each* beam bunch crossing; offline reconstruction/analysis apply event selections to select the interesting set of events for physics measeruements

▸ Option 1 for event key is the beam crossing counter

  ◦ GTU counting 98.5MHz beam crossing clock with a 64bit counter
  ◦ DAQ/electronics will broadcast EIC beam crossing counter to indexing all detector hits

▸ Option 2 for event key could be a tuple
(run, time-frame, crossing counter in time-frame)

▸ Either is sufficient. Could use both too

Reference to last meeting,
Nathan's talk [link]

Event key

- Generalizes the concept of event number and possibly run number to streaming scenarios
- Event number: For each level in the event hierarchy, have:
  - Absolute number: Starts at 0, increments by 1 monotonically
  - Relative number: Starts at 0 for each parent, increments by 1 monotonically
  - User key: Could be anything
- Run number:
  - Key for reloading resources such as calibrations
  - Helps to be a number, not an interval

Discussion 2: what is an (DAQ) run for ePIC?

This is a discussion. Scenarios for a "DAQ run" could be:

▸ Electron bunch replacement at O(1)Hz

   ◦ Restarted automatically driven by accelerator bunch replacement control

   ◦ Effectively a luminosity window, O(1000) ePIC time frames, require lumi/polarization measurement, scalar reading synchronized to the edge of the lumi window

▸ Data taking period between human-driven configuration changes (~1hr)

   ◦ Commonly used by many experiment, neatly mapped in configuration DB storage

▸ Entire hadron ring fill (few hours)

▸ Not using a DAQ run concept, just luminosity window/time frames

In any case, run start/end will be marked with beam crossing counter at GTU

Discussion 3: slow control (SC) data

- It is good practice to embed slow control data in raw data, but embedded data are hard to use
  - Some periodic reading require interpolation between readings (e.g. temperature); some requires future slow control reading (masking unstable FEEs in deadmap)
- Slow control data will be recorded to online DBs
  - Slow control recording persists regardless data taking
  - A mirror of online DB will be available for offline use
- *Suggest* detach slow control data access from reconstruction pass
  - Instead, use online database sources to produce calibration files (gain map, deadmap, etc.) as input to reconstruction, with validity marked with beam counter ranges
  - Use (automated) calibration job to process slow control data to form calibration input to reconstruction jobs, fits well in the multi-pass calibration computing plan
- Calibration access require scalable calibration database in offline world
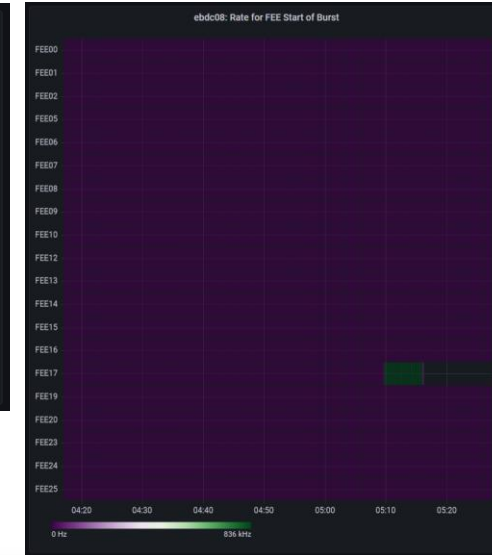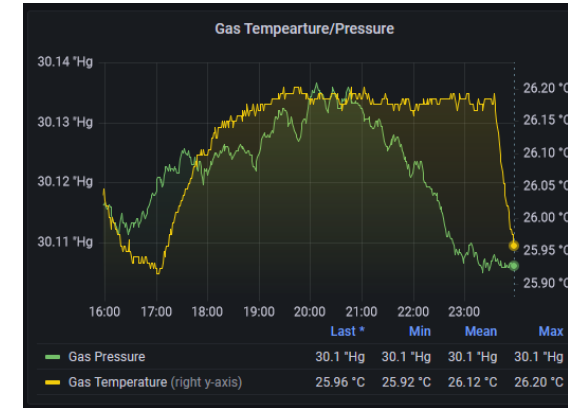
Reference to last meeting, Nathan's talk [link]
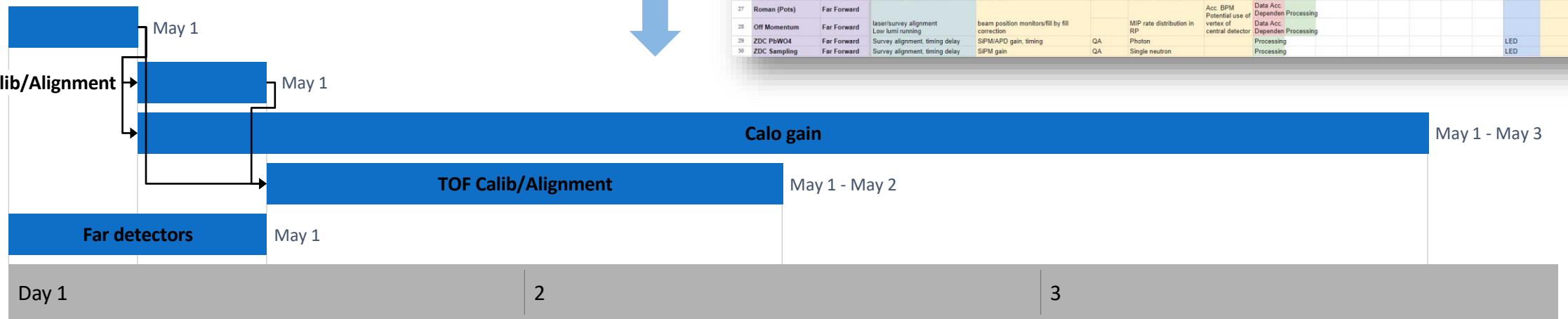

Memory management
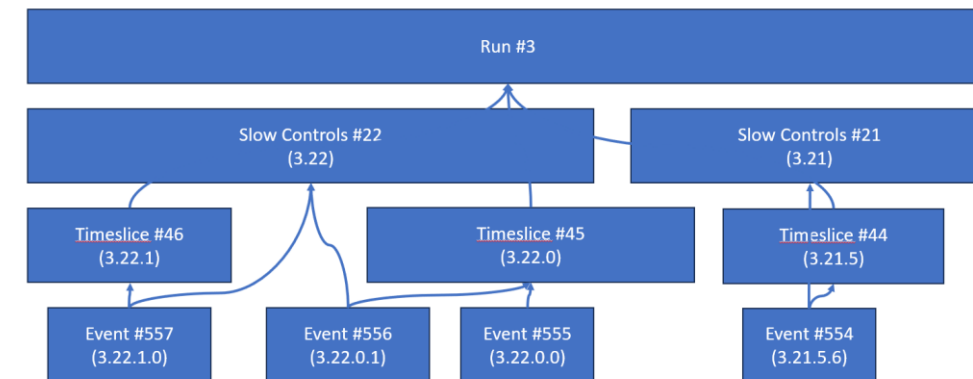
Discussion 4: Calibration workflow

- Calibration workflow seems fits into the prompt reconstruction computing model. Inputs welcomed.

- High level summary plot:

Extra Information

>> Feel free to share your views

Live note on indico [link]

Experiment Clock

- Clock will be distributed from GTU to FEB to synchronize digitizers and tag time of the hits
- For collider experiment, it is common to synchronize FEB clock to a harmonics of the beam collision clock
  - Absolute time of hit is not useful
  - But relative time to bunch crossing is critical for T0, spin, and luminosity tagging
  - EIC Clock frequency: 98.5MHz (no ramp variation), 1260 RF bunches, 12.8us/revolution
- SVT is a special case: fixed to LHC clock by lpGBT [40.078 MHz], slow [few -10us integrated], and synced to fast detectors offline [sPHENIX implementation]
- ePIC design specification discussion
  - We have multiple ASICs of various digitization frequency
    - E.g. ~40MHz (EICROC), ~50MHz (SALSA), ~200MHz (AstroPix)
  - Shall we distribute clock at 9.85MHz (1/10 harmonic of EIC crossing clock, 126*revolution frequency)?
    - Then FEB/DAM of each subsystem can generate their own synchronized clock at multiples of 9.85MHz
    - Existing example is sPHENIX 9.4MHz clock x 6*16bit per clock @ 1.1Gbps; W Gu tested to 7.9Gbps
  - Beam clock counter and sync signal broadcasted from GTU->DAM->FEE, and embedded in data stream

Example: sPHENIX clock data embedding at 6x 9.4MHz beam clock, 12Byte/beam clock [sPHENIX TDR]

clock count		0	1	2	3	4	5
bits 0-7	mode bits/BCO	mode bits	BCO bits 0-7	BCO bits 8-15	BCO bits 16-23	BCO bits 24-31	BCO bits 32-39
bit 8	beam clock	1	0	0	0	0	0
bit 9	LVL1 accept	X	0	0	0	0	0
bit 10	endat0	X	X	X	X	X	X
bit 11	endat1	X	X	X	X	X	X
bit 12	modebit en.	1	0	0	0	0	0
bits 13-15	3 user bits	0	1	2	3	4

ePIC streaming computing: online to offline

Echelon 0 computing at streaming readout DAQ

▸ Readout routing, time frame building [see Discussion 1]

▸ Primary function: data reduction
  ◦ Traditional DAQ: triggering was the main method of data reduction, assisted by high level triggering/reconstruction, compression
  ◦ Streaming DAQ need to reduce data computationally: zero-suppression, feature building, lossless/lossy compression

▸ Challenge: any information loss is permanent; observe full DAQ rate with less than O(1min) of latency
  ◦ Reliable data reduction methods; Sized to peak data rate + contingency; More expensive (than offline) to develop and maintain
  ◦ → Application, only if needed; three subsystem need identified below

▸ Other critical roles:
  ◦ Slow control; Monitoring (in coordination with monitoring via prompt reconstruction); Meta data collection, database service

Detector Group	Channels					RDO	Fiber	DAM	Data Volume (RDO) (Gb/s)	Data Volume (To Tape) (Gb/s)
	MAPS	AC-LGAD	SiPM/PMT	MPGD	HRPPD					
Tracking (MAPS)	36B					400	800	17	26	26
Tracking (MPGD)				202k		118	236	5	1	1
Calorimeters	500M		104k			451	1132	19	502	28
Far Forward	300M	2.6M	170k			178	492	8	15	8
Far Backward	82M		2k			50	100	4	150	1
PID (TOF)		7.8M				500	1500	17	31	1
PID Cherenkov			320k		140k	1283	2566	30	1275	32
TOTAL	36.9B	10.4M	596k	202k	140k	2980	6826	100	2,000	96

3 subsystem data reduction need beyond FEB/RDO zero-suppression

Calorimeter cluster building (CPU/GPU?)

FB high-rate tracker: Tracklet building (CPU/GPU?)
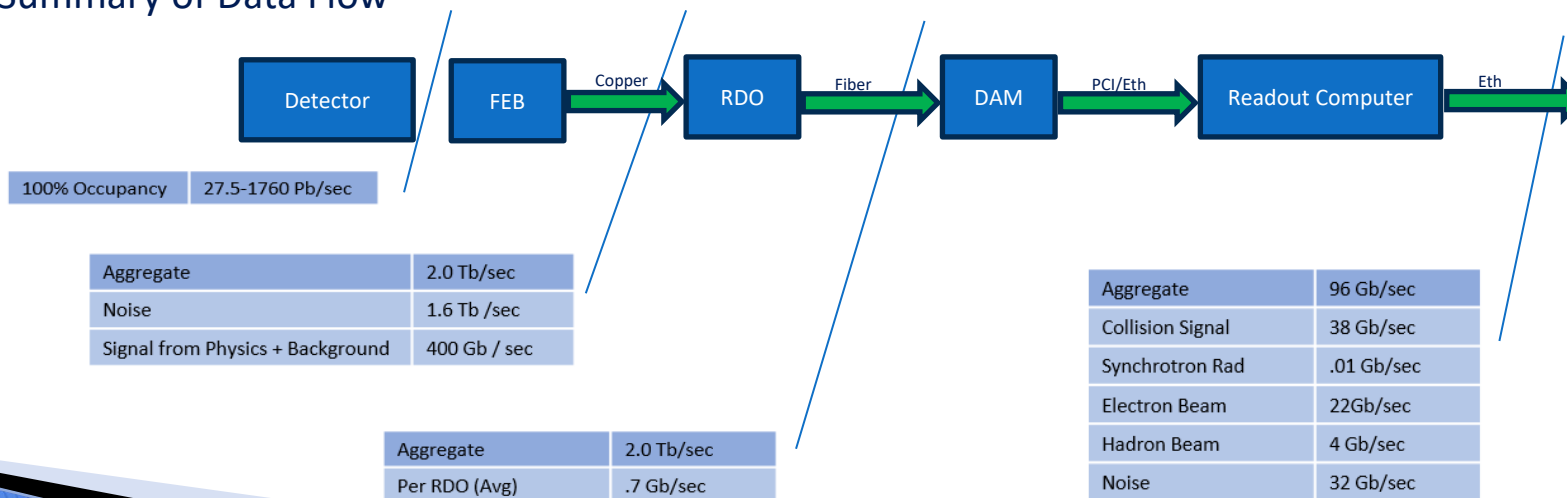
dRICH: Collision throttling (2 tier DAM FPGA)

EPIC Detector Scale and Technology Summary:

Detector System	Channels	RDO	Gb/s (RDO)	Gb/s (Tape)	DAM Boards	Readout Technology	Notes
Si Tracking: 3 vertex layers, 2 sagitta layers, 5 backward disks, 5 forward disks	7 m^2 36B pixels 5,200 MAPS sensors	400	26	26	17	MAPS: Several flavors: curved its-3 sensors for vertex Its-2 staves / w improvements	Fiber count limited by Artix Transceivers
MPGD tracking: Electron Endcap Hadron Endcap Inner Barrel Outer Barrel	16k 16k 30k 140k	8 8 30 72	1	.2	5	uRWELL / SALSA uRWELL / SALSA MicroMegas / SALSA uRWELL / SALSA	64 Channels/Salsa, up to 8 Salsa / FEB&RDO 256 ch/FEB for MM 512 ch/FEB for uRWELL
Forward Calorimeters: LFHCAL HCAL insert* ECAL W/SciFi Barrel Calorimeters: HCAL ECAL SciFi/PB ECAL ASTROPIX Backward Calorimeters: NHCAL ECAL (PWO)	63,280 8k 16,000 7680 5,760 500M pixels 3,256 2852	74 9 64 9 32 230 18 12	502	28	19	SiPM / HG2CROC SiPM / HG2CROC SiPM / Discrete SiPM / HG2CROC SiPM / HG2CROC Astropix SiPM / HG2CROC SiPM / Discrete	Assume HGCROC 56 ch * 16 ASIC/RDO = 896 ch/RDO 32 ch/FEB, 16 FEB/RDO estimate, 8 FEB/RDO conserve. HCAL 1536x5 *HCAL insert not in baseline Assume similar structure to its-2 but with sensors with 250k pixels for RDO calculation. 24 ch/feb, 8 RDO estimate, 23 RDO conservative
Far Forward: B0: 3 MAPS layers 1 or 2 AC-LGAD layer 2 Roman Pots 2 Off Momentum ZDC: Crystal Calorimeter 32 Silicon pad layer 4 silicon pixel layers 2 boxes scintillator	300M pixel 1M 1M (4 x 135k layers x 2 dets) 640k (4 x 80k layers x 2 dets) 400 11,520 160k 72	10 30 64 42 10 10 10 2	15	8	8	MAPS AC-LGAG / EICROC AC-LGAD / EICROC AC-LGAD / EICROC APD HGCROC as per ALICE FoCal-E	3x20cmx20cm 600^cm layers (1 or 2 layers) 13 x 26cm layers 9.6 x 22.4cm layers There are alternatives for AC-LGAD using MAPS and low channel count DC-LGAD timing layers
Far Backward: Low Q Tagger 1 Low Q Tagger 2 Low Q Tagger 1+2 Cal 2 x Lumi PS Calorimeter Lumi PS tracker	1.3M pixels 480k pixels 700 1425/75 80M pixels	12 12 1 1 24	150	1	4	Timepix4 Timepix4 (SiPM/HG2CROC) / (PMT/FLASH) Timepix4	
PID-TOF: Barrel Endcap	2.2M 5.6 M	288 212	31	1	17	AC-LGAD / EICROC (strip) AC-LGAD / EICROC (pixel)	bTOF 128 ch/ASIC, 64 ASIC/RDO eTOF 1024 pixel/ASIC, 24-48 ASIC/RDO (41 ave)
PID-Cherenkov: dRICH pfRICH DIRC	317,952 69,632 69,632	1242 17 24	1240 24 11	13.5 12.5 6	28 1 1	SiPM / ALCOR HRPPD / EICROC (strip or pixel) HRPPD / EICROC (strip or pixel)	Worse case after radiation. Includes 30% timing window. Requires further data volume reduction software trigger

Summary of Channel Counts

Detector Group	Channels					RDO	Fiber	DAM	Data Volume (RDO) (Gb/s)	Data Volume (To Tape) (Gb/s)
	MAPS	AC-LGAD	SiPM/PMT	MPGD	HRPPD					
Tracking (MAPS)	36B					400	800	17	26	26
Tracking (MPGD)				202k		118	236	5	1	1
Calorimeters	500M		104k			451	1132	19	502	28
Far Forward	300M	2.6M	170k			178	492	8	15	8
Far Backward	82M		2k			50	100	4	150	1
PID (TOF)		7.8M				500	1500	17	31	1
PID Cherenkov			320k		140k	1283	2566	30	1275	32
TOTAL	36.9B	10.4M	596k	202k	140k	2980	6826	100	2,000	96

Summary of Data Flow

| Detector | → FEB | →(Copper)→ RDO | →(Fiber)→ DAM | →(PCI/Eth)→ Readout Computer | →(Eth)→ |

100% Occupancy	27.5-1760 Pb/sec

Aggregate	2.0 Tb/sec
Noise	1.6 Tb /sec
Signal from Physics + Background	400 Gb / sec

Aggregate	2.0 Tb/sec
Per RDO (Avg)	.7 Gb/sec

| Aggregate	96 Gb/sec
Collision Signal	38 Gb/sec
Synchrotron Rad	.01 Gb/sec
Electron Beam	22Gb/sec
Hadron Beam	4 Gb/sec
Noise	32 Gb/sec

Streaming DAQ – Computing : consideration 1
For kickstart the discussion, please interrupt to discuss at any moment

- Streaming DAQ naturally leads to no clear separation of streaming DAQ and computing
  - Streaming DAQ relies on data reduction computationally (i.e. no real-time triggering) → Any data reduction in streaming DAQ is a computing job
  - Which could be done at ASIC, FPGA, online-computers
  - Example could be zero-suppression (simple or sophisticated), feature extraction (e.g. amplitude in calo and tracklet in FB tracker)
  - Require minimal loss of collision signal; any data reduction require stringent bias control/study
- Citing ePIC software principles https://eic.github.io/activities/principles.html :
  We will have an unprecedented compute-detector integration:
  - We will have a common software stack for online and offline software, including the processing of streamed data and its time-ordered structure.
  - We aim for autonomous alignment and calibration.
  - We aim for a rapid, near-real-time turnaround of the raw data to online and offline productions.
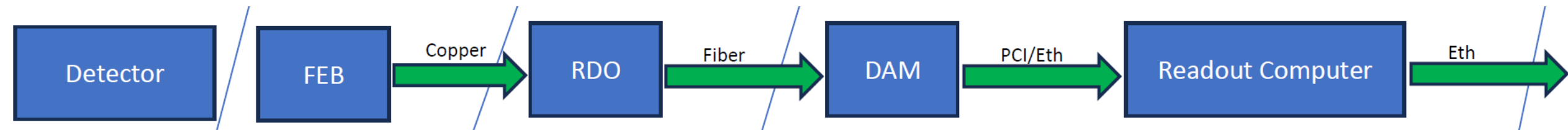
Streaming DAQ – Computing : consideration 2
For kickstart the discussion, please interrupt to discuss at any moment

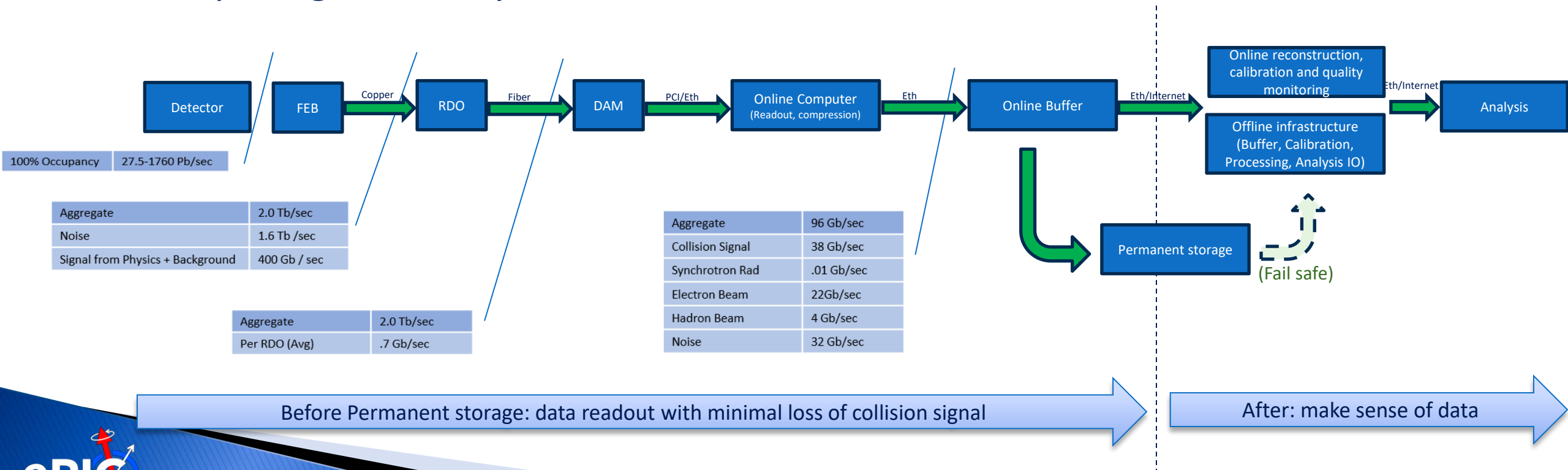▸ Sooner or later, a copy of data is stored and saved for permanent storage

▸ This stage of first permanent storage could be viewed as a DAQ – computing boundary

Detector	FEB	Copper	RDO	Fiber	DAM	PCI/Eth	Online Computer (Readout, compression)	Eth	Online Buffer	Eth/Internet

100% Occupancy	27.5-1760 Pb/sec

Aggregate	2.0 Tb/sec
Noise	1.6 Tb /sec
Signal from Physics + Background	400 Gb / sec

Aggregate	2.0 Tb/sec
Per RDO (Avg)	.7 Gb/sec

Aggregate	96 Gb/sec
Collision Signal	38 Gb/sec
Synchrotron Rad	.01 Gb/sec
Electron Beam	22Gb/sec
Hadron Beam	4 Gb/sec
Noise	32 Gb/sec

Online reconstruction, calibration and quality monitoring

Offline infrastructure (Buffer, Calibration, Processing, Analysis IO)

Eth/Internet

Analysis

Permanent storage

(Fail safe)

Before Permanent storage: data readout with minimal loss of collision signal

After: make sense of data

Streaming DAQ – Computing : consideration 2
For kickstart the discussion, please interrupt to discuss at any moment

- Paid by project
- Has a hard archival limit (O(100Gbps)) from both throughput and tape cost
- Main goal on "online-computing" is data reduction to fit output pipeline
- Stringent quality and bias control for any lossydata reduction
- As minimal reduction as affordable to
  ◦ (1) reduce unrecoverable systematic uncertainty
  ◦ (2) reduce complexity, cost, failure modes.
  ◦ Any processing beyond minimal need a physics motivation to justify project cost/schedule reviews (and possible descope reviews)
- High availability: any down time cost $O(0.1)M/day → usually on host lab

- Driven by collaboration, operation fund
- We would like to complete within a small latency (<O(1)week)
  ◦ Usually driven by calibration and debugs
- Main goal on "offline-computing" is to bring out physics objects for analysis
- Quality control for reconstruction
- Can afford to redo reconstruction if new algorithm or with new physics insights (at cost of time, effort and computing)
- Can wait for short interruptions and can be distributed

Before permanent archival: DAQ	After permanent archival: Computing

Why streaming DAQ/computing?

	EIC	RHIC	LHC → HL-LHC
Collision species	$\vec{e} + \vec{p}, \vec{e} + A$	$\vec{p} + \vec{p}/A, A + A$	$p + p/A, A + A$
Top x-N C.M. energy	140 GeV	510 GeV	13 TeV
Bunch spacing	10 ns	100 ns	25 ns
Peak x-N luminosity	10^{34} cm^{-2} s^{-1}	10^{32} cm^{-2} s^{-1}	$10^{34} \rightarrow 10^{35}$ cm^{-2} s^{-1}
x-N cross section	**50 µb**	**40 mb**	**80 mb**
Top collision rate	500 kHz	10 MHz	1-6 GHz
$dN_{ch}/d\eta$ in p+p/e+p	0.1-Few	~3	~6
Charged particle rate	**4M N_{ch}/s**	**60M N_{ch}/s**	**30G+ N_{ch}/s**

▸ Events are precious and have diverse topology → hard to trigger on all process

▸ Signal data rate is moderate → possible to streaming recording all collision signal, event selection in offline reconstruction using all detector information after calibration

▸ Background and systematic control is crucial → avoiding a trigger bias; reliable data reduction