Reconstruction & Data Management Model

Matt Graham

HPS Software Meeting

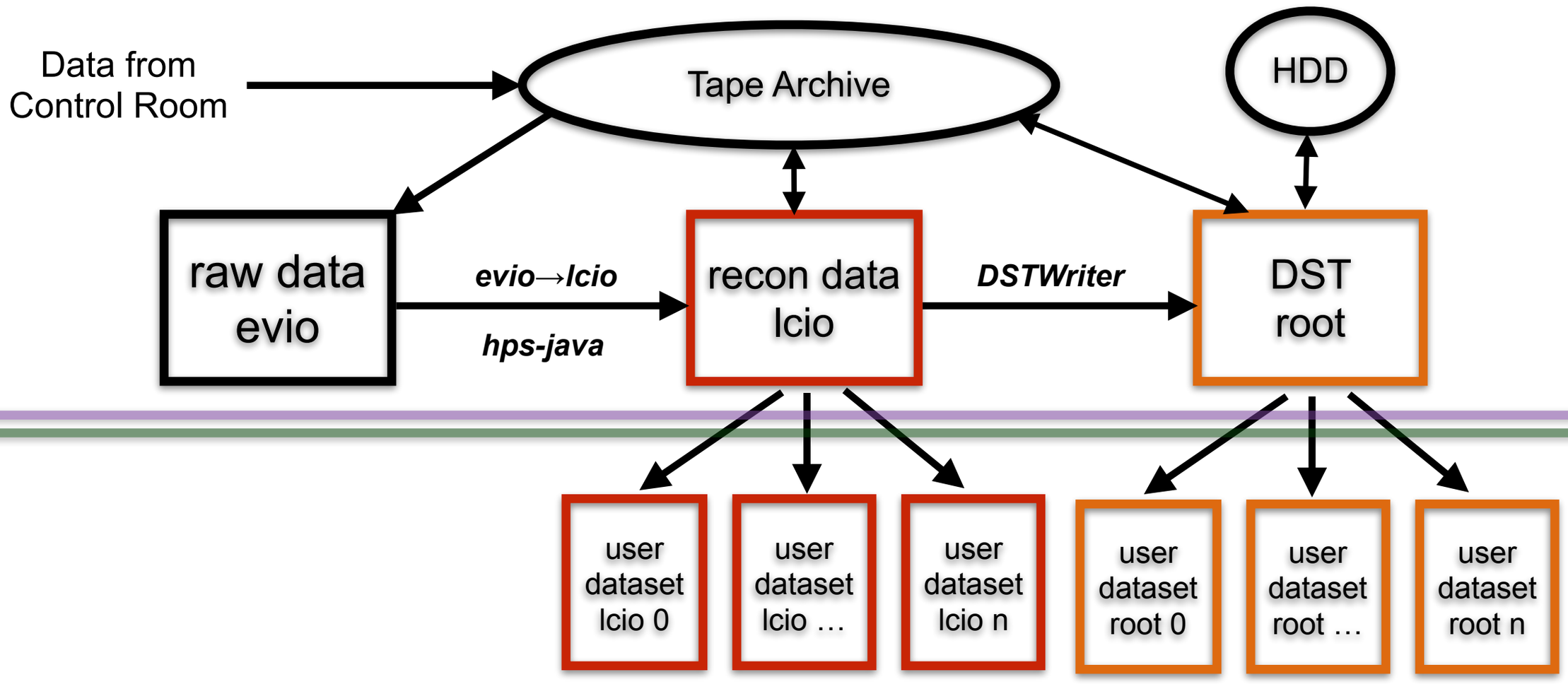
August 7, 2014

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

current production & management model

Central production→all events saved in evio/recon/dst formats



User production→select samples of interest from either recon-lcio or DST
(somewhat users choice though recon-lcio has more info);
shouldn't have to go back to the raw evio ever

Offline computing requirements: Data storage

# events/week	5.2 E 9	
raw event size	3.5 kB	
raw event storage	***16 TB***	
recon event size	15 kB	
recon event storage	***69 TB***	***~4x raw***
DST event size	2.6 kB	
DST event storage	***12 TB***	
Total storage/week	***97 TB***	

Standard assumptions: 1 week, 200 nA @ 2.2 GeV;
trigger rate = 8.6 kHz

This was not what was shown at the experimental readiness review!
We used a 15.8kHz trigger (everything scaled by 15.8/8.6)

Offline computing requirements: Data processing

# events/week	5.2 E 9	
reco time/event	55 ms	expected, with 8 ns cut
total cpu time/week	***3.3k cpu-days***	
recon evt/job	~570k	2.2 GB files
# of batch jobs	9.1k	
cpu time/job	8.7 hours	
total wall time	***~7 days***	***assume 500 batch slots***

Standard assumptions: 1 week, 200 nA @ 2.2 GeV;
trigger rate = 8.6 kHz
DST & data quality are very fast…add ε to the total

Why is the recon 4x the raw data? We save a lot of stuff.

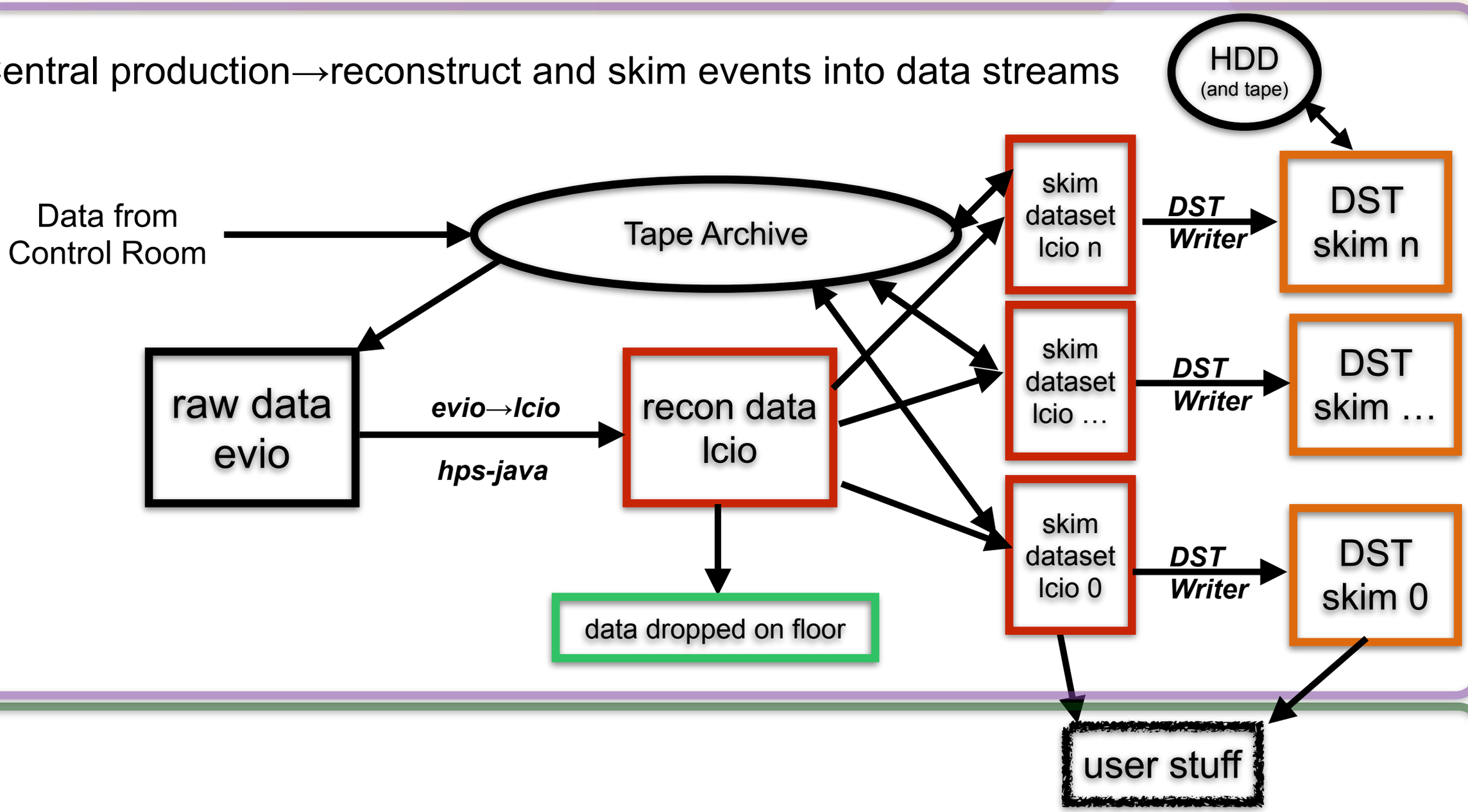
Collection Name	Type	Class Where Filled	Description
EcalReadoutHits	RawCalorimeterHit	evio.ECalEvioReader (data)	ECal Hits in ADC counts
EcalCalHits	CalorimeterHit	recon.ecal.EcalRawConverterDriver	Calibrated ECal Hits
EcalClusters	Cluster	recon.ecal.EcalClusterICBasic	ECal Clusters
SVTRawTrackerHits	RawTrackerHit	evio.ECalEvioReader (data)	Si sensor single strip hit information
SVTShapeFitParameters	GenericObject	recon.tracking.RawTrackerHitFitterDriver	result of the ADC vs sample # fit
SVTFittedRawTrackerHits	LCRelation	recon.tracking.RawTrackerHitFitterDriver	relation between SVTRawTrackerHits and SVTShapeFitParameters
StripClusterer_SiTrackerHitStrip1D	TrackerHit	recon.tracking.DataTrackerHitDriver	1D Si strip clusters
HelicalTrackHits	TrackerHit	recon.tracking.HelicalTrackHitDriver	3D SVT hits combining StripClusterer_SiTrackerHitStrip1D hits in axial/stereo layers
HelicalTrackHitRelations	LCRelation	recon.tracking.HelicalTrackHitDriver	relation between HelicalTrackHits and StripClusterer_SiTrackerHitStrip1D
HelicalTrackHitMCRelations	LCRelation	recon.tracking.HelicalTrackHitDriver	relation between HelicalTrackHits and MCParticles
RotatedHelicalTrackHits	TrackerHit	recon.tracking.HelicalTrackHitDriver	same as HelicalTrackHits but rotated into SeedTracker tracking frame (x->y,y->z,z->x)
RotatedHelicalTrackHitRelations	LCRelation	recon.tracking.HelicalTrackHitDriver	.... but rotated into SeedTracker tracking frame (x->y,y->z,z->x)
RotatedHelicalTrackHitMCRelations	LCRelation	recon.tracking.HelicalTrackHitDriver	.... but rotated into SeedTracker tracking frame (x->y,y->z,z->x)
MatchedTracks	Track	recon.tracking.TrackerReconDriver	tracks found in the SVT with (d0,phi,omega,tanlambda,z0) parameters
FinalStateParticles	ReconstructedParticle	recon.particle.HpsReconParticleDriver	the list of final state particles (electrons, positrons, photons) with 4-momenta.
UnconstrainedV0Candidates	ReconstructedParticle	recon.particle.HpsReconParticleDriver	electron-positron pairs with vertex (unconstrained)
BeamspotConstrainedV0Candidates	ReconstructedParticle	recon.particle.HpsReconParticleDriver	electron-positron pairs with vertex/momentum required to point back to beamspot at target
TargetConstrainedV0Candidates	ReconstructedParticle	recon.particle.HpsReconParticleDriver	electron-positron pairs with the vertex z fixed to the target position and the (x,y) constrained to beamspot
FPGAData	GenericObject		SVT hybrid information (e.g. temperature)...anything else?
TriggerBank	GenericObject		trigger information for the event
ReadoutTimestamps	GenericObject		event timestamp

…this is by design…point was to make the recon the main repository of event information. There may be some redundancy and/or chafe (2 sets of HTH, etc), but not much. Dropping collections will involve dropping information…it's a trade off

Includes raw hits (i.e. ADC counts)→redundant with evio … I think useful but it could go

alternative production & management model

Central production→reconstruct and skim events into data streams



User production→select sub-samples from skimmed samples if wanted

Benefits & drawbacks of the skimmed approach

- Benefits
  - probably saves on tape/disk space (as long as the skims define sufficiently independent samples)
  - re-running *analysis* over a skimmed recon-lcio sample should be much quicker
    - I worry about time taken to queue/mount ***all*** recon-lcio events…may dominate the "analysis time" by large factor
      - you probably wouldn't run reconstruction just on the skimmed data…event can migrate in & out of skims
- Drawbacks
  - defining a skim is quite a bit of work
  - implementing a skim into production is some work
  - bookkeeping skims is quite a bit of work
  - will definitely add more production cycles to the lifetime of the experiment→more processing (for sure) and possibly even more storage over the long term

Takeaway

- For the first data, the original model is the way to go…we want to have every recon event and as much info saved as possible so we can fix problems. NOBODY DISAGREES WITH THIS.
- Beyond that, the skimming model may be the way to go (or something else, similar). Frequent production cycles are a good thing; skims bring the datasets closer to what individual users want, etc etc
- They are a lot of work to develop & maintain though…