Plans for data processing in 2020c/2021

36th B2GM, DP session

23/06/2020

Stefano Lacaprara, Marco Milesi

INFN Padova, Uni Melbourne

Summary

- Status since February 2020 B2GM/BPAC
- Plan for 2020c, early 2021 prompt processings
- Plan for full reprocessing of ~70 /fb (“proc11++”, 2019+2020a)

Reminder of current data processing flow

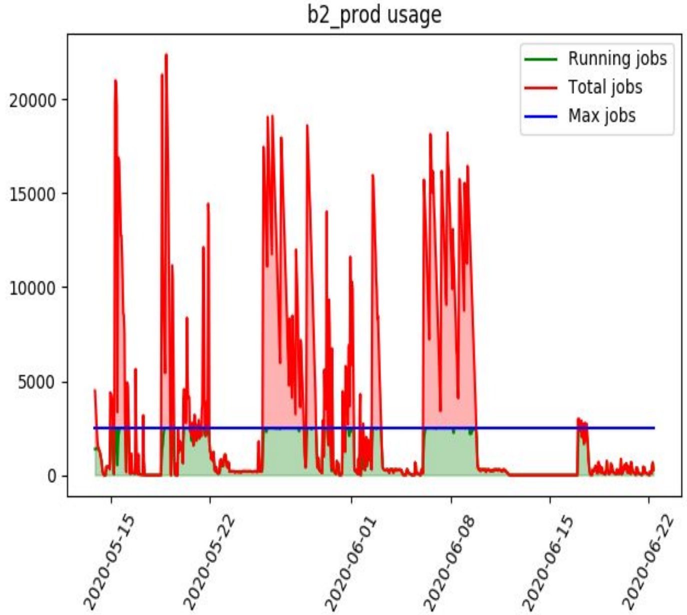
- Unofficial
  - Run as soon as RAW lands on dataprod/, using conditions of online GT
  - I/O:
    - (For Mirabelle) I: hlt_hadron, O: cdst + offskim
- Prompt (bucketXX)
  - First processing after automated (Airflow) calibration → mdst
  - In steady state, ~10 fb^{-1} / bucket
    - now ~3/4 weeks, will be ~1 week of data taking
- Official (procXX)
  - Complete (re)processing of data → mdst
    - @KEKCC for HLT skims
    - On the grid (BNL, KEK) for all events

Post-mortem of 2019 (proc11), 2020a (prompt)

- KEKCC resources (b2_prod) bumped up to 2500 cores
- Data taking campaigns (mostly) run in HLT "monitoring" mode → any event is processed on the grid

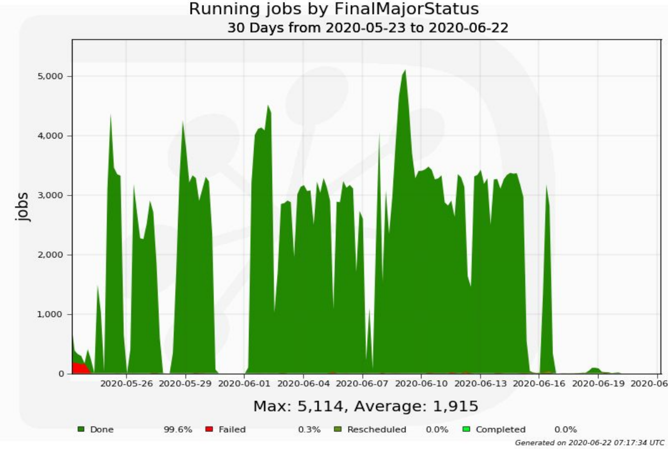
	∫L dt [/fb]	ΔT [d] - local (HLT skims)	ΔT [d] - grid (all)
proc11	8.7	4	15
bucket9	2.7	8 (*)	3 (+7*)
bucket10	10.4	4	17
bucket11	12.7	4	14
bucket12	2.4	1	15

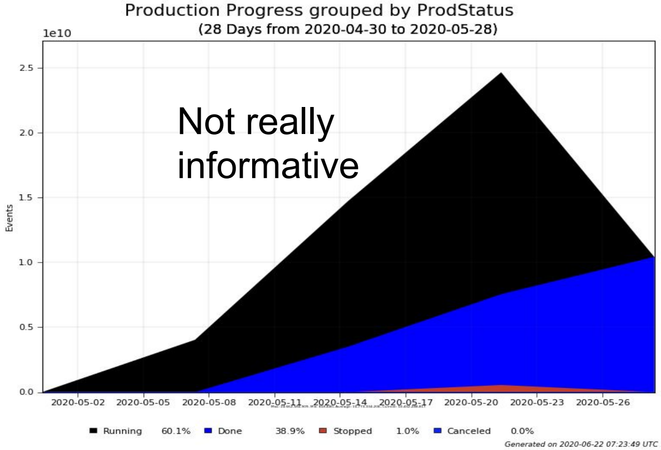
(*) missing runs had to be re-submitted



Grid post-mortem - 2019+2020a processing

- Staging data well in advance: key for success
  - Now manual as the unstage: aim for smart automation
- Good job of DP shifters for RawProcessing
  - Early discovery of off-res GT payload missing in proc11
- “Babysitting” by experts is time consuming
  - Need more DP-tailored CLI and DIRAC tools to improve productivity (also for non-experts).
    - Eg, *status by campaign vs time*, for RawProc and RawMerge, separately
    - Quickly identify “true” crashes (e.g., basf2, CDB):
      - We have b2dp-monitor-grid which parses gb2_prod_summary: can do that “natively”?
      - In these cases, we cancel the input file from production: need to keep track/recover. How?





Not really informative

Room for improvement - grid production

- Merge step is often the real bottleneck
  - Can be longer than actual processing!
  - Long tail in total processing time b/c last few % of merge fabrications.
    - **Can we envisage to perform the merge step at the same site as the processing step?**
- Optimisation of ProdID size
  - Now we have 100 runs/ProdID, but run size (in fb^{-1}) is variable, no guarantee to have good splitting
    - The larger the ProdID, the longer to complete
    - Analysers need to access files scattered over many ProdIDs: not ideal.

Plans for 2020c (and beyond)

- Drop unofficial processing:
  - Mirabelle offskim production to be moved in express reco/online
- Drop local processing at KEKCC:
  - Not clear how many dedicated resources we will effectively have after summer...
- Prompt + official grid processing:
  - What to process and in which priority
  - Integration of udst production (analysis+systematic skims) in processing

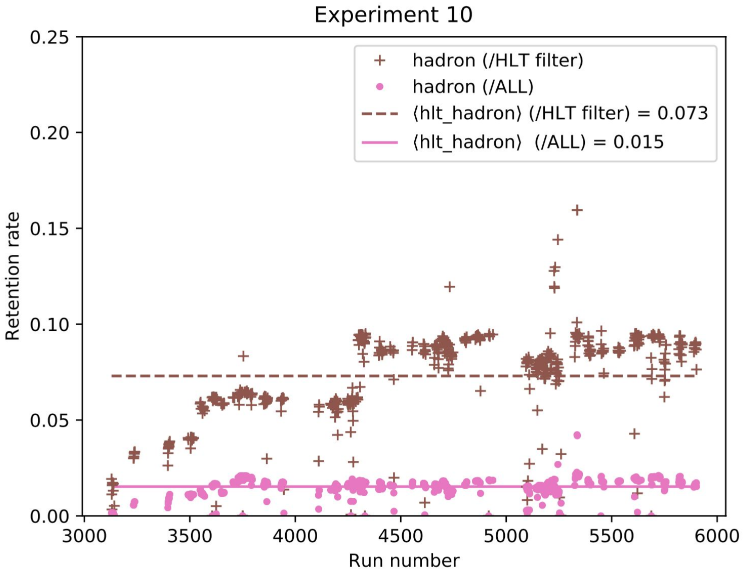
RAW data inputs and “physics streams”

- So far, B-physics done on hlt_hadron skim
  - Originally introduced for calibration (alongside other HLT skims)
  - Event flag defined *online* at HLT level:

```
[[nTracksLE>=3] (*) and [Bhabha2Trk==0]]
```

  - Retention rate in data: ~2% (/all events), 10% (/hlt-filtered events)
  - Fast sampling of RAW hlt_hadron-*only* data (CC): smaller inputs to processing.
    - RAW skimmed data replicated on grid SEs
- Tacit assumption: 100% efficient on data and MC for typical offline analyses selections.
  - *(Analysts *should* study hlt_hadron efficiency with high priority → use 2020a grid mdsts, no HLT filter!)*

(*) pT > 0.2 & abs(d0) < 2 & abs(z0) < 4

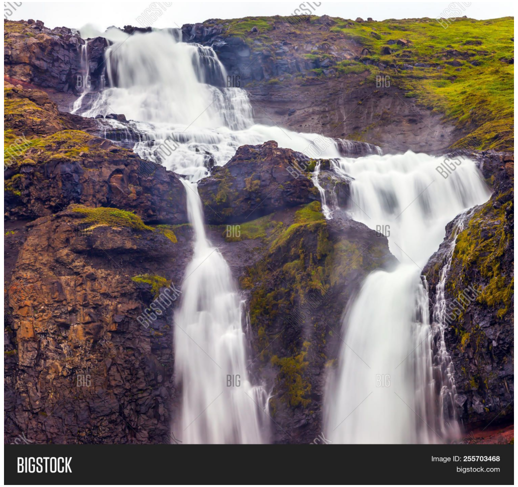


RAW data inputs and “physics streams”

- Different HLT-skimmed RAW data can be thought as “streams”
  - hlt_hadron skim → B-physics stream
  - hlt_* skim → *-physics stream
  - hlt_bhabhaecl (prescaled?) → offline luminosity stream

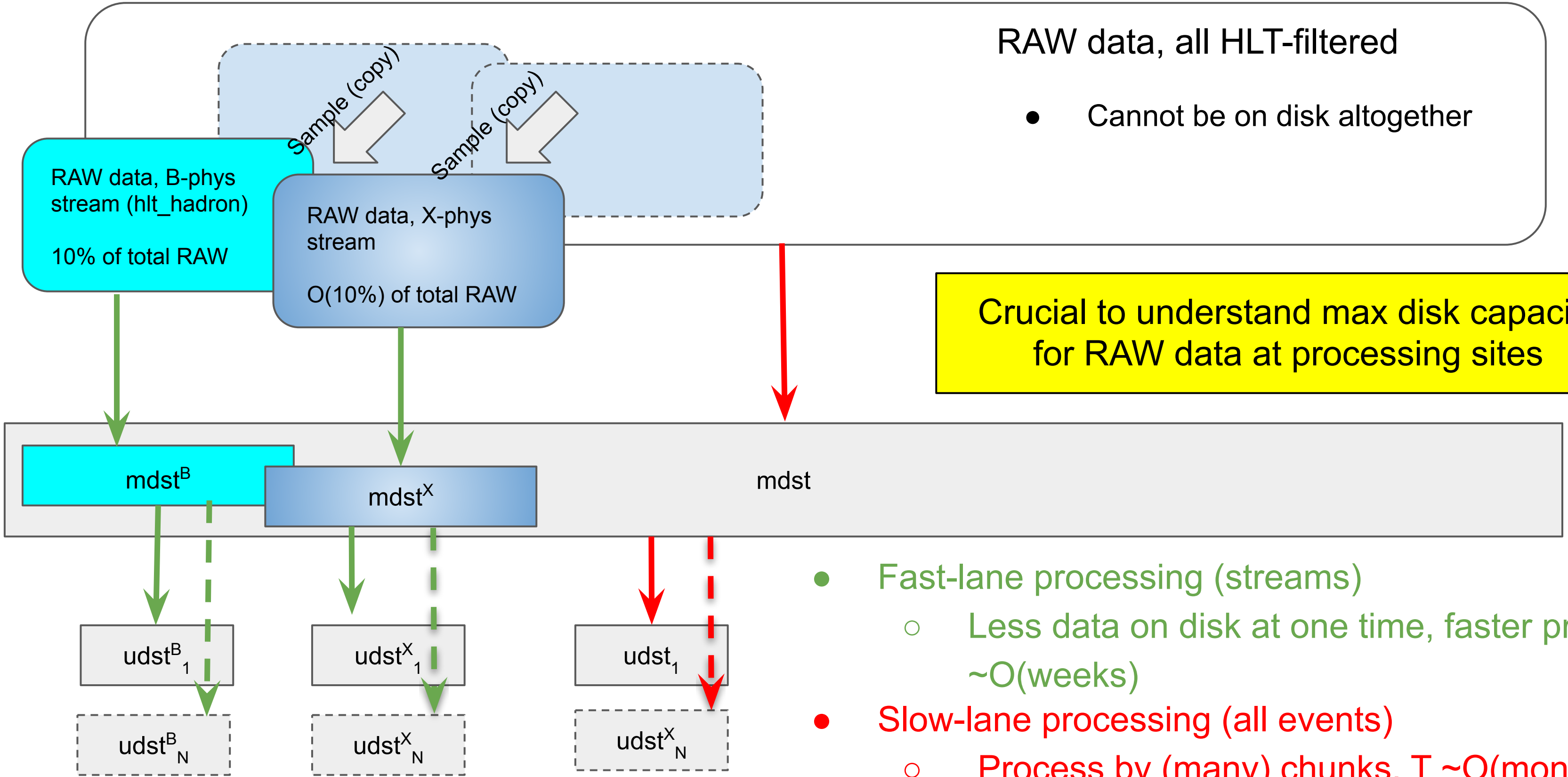
Why should we sample RAW dataset “streams” out of all data?

- Pro: fastest lane for processing
  - (up to) x% only of events to reconstruct
  - (up to) x% only of RAW data to stage on disk per processing
    - Much less stress on disk/tape resources
- Con: RAW data duplication
  - RAW “all” data must still be processed for non-B-physics:
    - DM, taus, long lived particles, magnetic monopoles…
    - Performance studies (e.g., lepton ID)



Scheme proposal for stream-based processing

For a given processing campaign (prompt, official):



- Cannot be on disk altogether

Crucial to understand max disk capacity for RAW data at processing sites

- Fast-lane processing (streams)
  - Less data on disk at one time, faster processing: ~O(weeks)
- Slow-lane processing (all events)
  - Process by (many) chunks, T ~O(months)

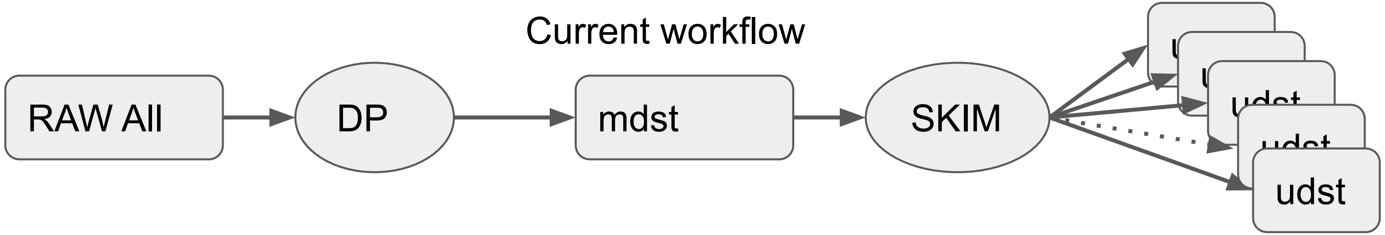
Caveats and thoughts

- **HLT skims originally designed for *calibration*:**
  - Not necessarily an exact subset of HLT trigger menu (aka, hlt_filter line). Different prescales, looser selection…
    - RAW HLT skims for calibration likely heavily pre-scaled at CC level in the near future...
  - If (some) HLT skims to become physics streams, *should* be upgraded to HLT trigger menu
- **Several streams to cover for more physics/performance use-cases?**
  - Some key points:
    - Non-proliferation policy → avoid too much RAW data duplication
    - Must be ~orthogonal w/ each other
    - “Stream selection efficiency” must be studied by analysers
- **Corner-cases will surely remain non-coverable by streams → need processing of “all” events**

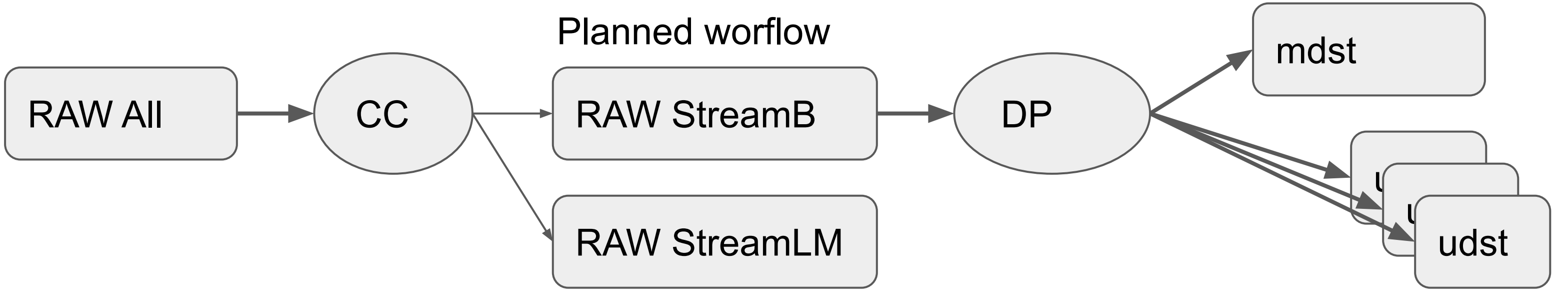
uDST (aka analysis skim)

- Currently run after mdst production is complete
  - Ready **way after** mdst are done: hard to be used in timely fashion by analysis
- Ongoing plan:
  - Produce udst(s) alongside mdst for hlt_hadron stream in same production
  - To test locally/grid in bucket13

Current workflow



Planned worflow



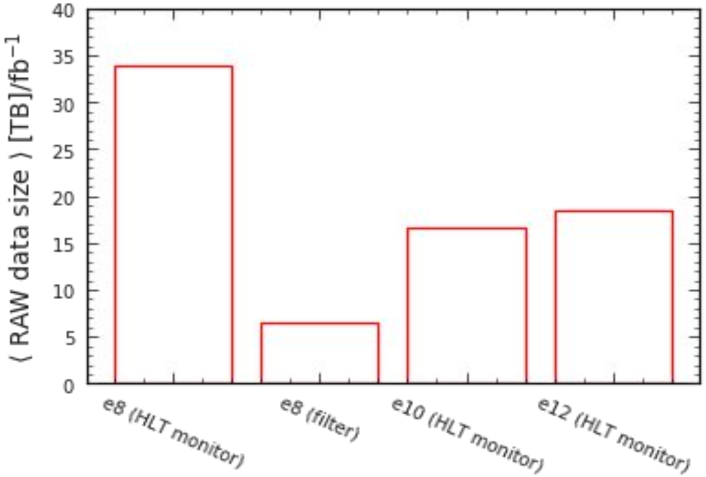
Caveats and thoughts

- Are udst actually ok for analysis?
  - WG liaisons should communicate specific requirements
- Which (and how many) udst to be produced?
  - Proposed workflow adds another step of processing → might not scale well on larger datasets
  - Merge step
    - Often the bottleneck of production on grid
    - If multiple output file, multiple merge. Further problems?
  - First feedback from DC: up to 10 udst might be ok, more can be problematic
- Mdst and udst have different size: merging to target size to be tested
- Will start with just one udst (systematics skim) and learn from experience
- ...

Resource estimate: prompt processing

Summary of resource estimate (assuming steady state, no backlog) for prompt → 1 bucket/week, ~10/fb / week

- **HLT_monitor mode**
  - Disk needs:
    - Estimated 20 TB RAW data / fb-1, 200 TB per week
    - If keeping 3-4 buckets alive at one time : about 6-800 TB of “live” data on disk in FIFO mode
  - CPU needs (based on 2020a prompt processing):
    - ~4k jobs max → 15 days / 10fb^{-1}
    - WARNING: 2 weeks to process 1 week of data
- **HLT_filter mode**
  - **Disk : ~20% : 150 TB alive on disk at one time**
  - **CPU: ~50% : <2k CPU> + merging**



Resource estimate for proc11++, O(100) /fb

- Based on proc11, estimate disk input and output, CPU and time with current BNL resources
- proc11 done on grid in 15 days: L=~10 /fb
    - All events, no HLT filtering
- **proc11++ 2019+2020a (?): L ~70 /fb** (release-05?)
    - Mostly (as of today) with HLT in monitoring
    - Extrapolating:
        - all events: 7*15=100 days: **3.5 months**
        - hlt_hadron: **~1 week**

Resource estimate for proc11++, O(100) /fb

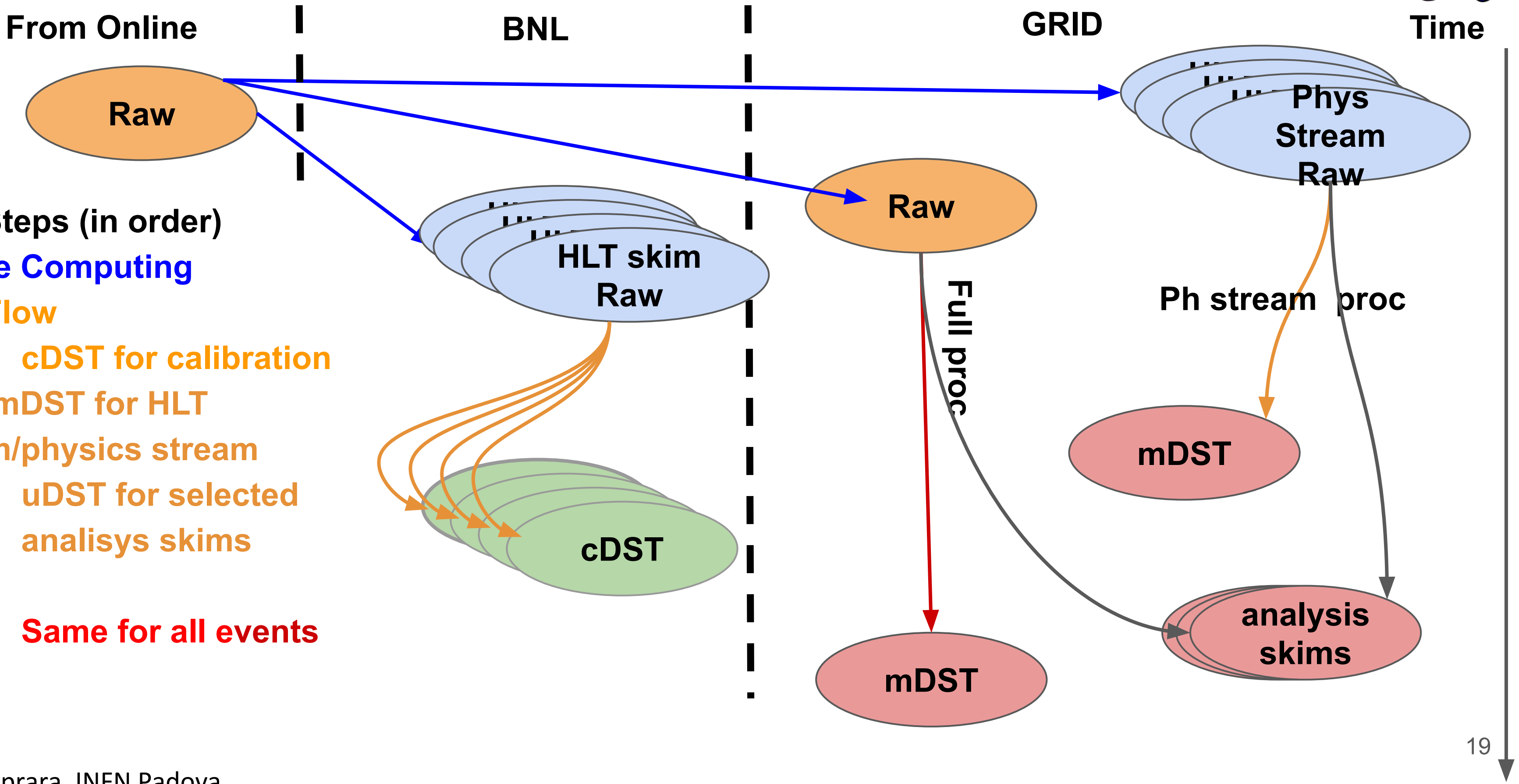
- Based on proc11, estimate disk input and output, CPU and time with current BNL resources
- proc11 done on grid in 15 days: L=~10 /fb
  - All events, no HLT filtering
- **proc11++ 2019+2020a (?): L ~70 /fb** (release-05?)
  - Mostly (as of today) with HLT in monitoring
  - Extrapolating:
    - all events: 7*15=100 days: **3.5 months**
    - hlt_hadron: **~1 week**

Miscellanea

- Offline luminosity
  - Will be no longer doable at KEKCC locally
  - Will need to test analysis on a dedicated stream on the grid
- Offline lumi now in txt files on confluence (then ported to sqlite DB by DP)
  - Need to upload to RunDB
    - tools/procedure to be developed Some preliminary instruction if you are interested in helping

BACKUP

Data Processing schema (plan)



Steps (in order)

- **Core Computing**
- **AirFlow**
  a. **cDST for calibration**
- **DP mDST for HLT skim/physics stream**
  a. **uDST for selected analisys skims**
- **DP:**
  a. **Same for all events**