Analysis Mini-Workshop Notebook, 2021-05-04

The usual live notes to record the discussion and follow-up points. Feel free to write down all observations and questions. Please tag comments with your name.

CMS

- Mariarosaria D'Alfonso is presenting for CMS
- Axel very useful data. More samples and datasets will increase the diversity of the data. Do you see that? While CMS will produce many, analysis will use a subset of them. Is it really the columnar model or just the speed of IO? A: Columnar data formats help with machine learning techniques.
- Lindsey: people like having a lightweight framework that is common among analysis groups.
- Lucas: In reduced format Nano do you include systematic uncertainties? A: no the central value is there but global quantities are used for uncertainties: JoshBD adds that pdf weights are saved.

ATLAS

- Johannes Elmsheuser presenting for ATLAS
- Josh BD: Can the special use case (12) be generalized? By making the smaller data formats a little fatter but specialized for the analysis? A: Absolutely if you know what extra is needed ahead of time. Low pT tracks are challenging in high volume datasets.
- Axel: Is PHYSLITE produced by ATHENA? Yes, then the analysis that uses it can be ROOT but not recommended, the lightweight framework is.
  ATLAS has enormous duplication in its DAOD format, James this is particularly a problem in MC datasets since it is physics rich… a single event can be written out >80 times.
- There is also extra information that could be optimized in format
- Axel: will reduced formats work better for HL-LHC? A. Yes, because we now know more what analysts need and the format is better targeted.

LHCb

- Eduardo Rodrigues presenting for LHCb
- Axel: We are giving up on the idea that analysis trains people in C++, how should we address this? A: push as a community and create a career path the motivations are stronger than ever (as computing becomes more heterogeneous) there are RSEs contributing now, we need more.
- Analysis users don’t need deep expertise

- Lindsey: using python wrappers is a good strategy for separating concerns between RSE work and physics work. Lindsey: strong compiler tool chains is needed but the language itself is not the issue
- Lukas: The other parts of our workflows still need C++; analysis can not be the only way to pickup workforce for that part. A: yes maintenance of the C++ parts is still a concern
- Nick: C++ is evolving every 3 years and is becoming more pythonic… we should concentrate on the algorithms
- Lindsey: in the past what we called analysis we now call reconstruction, we need to find people that are interested in the computing problems we face, they appear naturally A: yes, especially for the trigger

ALICE

- Peter Hristov presenting for ALICE
- Axel: Is it possible for people to write out nTupes, A: yes, slide 20 and then can use R-nTuple, Axel: Do you expect people to drive towards nTuples? A: ALICE has a long history of centralized analysis so this is not expected...skims are either really small, fitting on private resources (not GRID) or are histograms. Train model is still active they run every night of the order of 100 run (not every night though)
- Lukas: What does the average student use? Do they create trains or just use the outputs? A: 50/50 some just run on skims but some need more, What analysis system is used on the laptop? A: Flow? Jan says you would use the same framework locally because of data organization (associations) Students have low barrier to entries for trains QA is minimal they use an automatic staged approach with fractions of the full samples running first.

Pythonic Data Science

- Jim Pivarski presenting
- Doug: what is the scale of the distribution? 10s of datacenters? Less? A: Dask has been tested at 10Kcores, not big for us. ServiceX/Coffee is targeting for larger with larger numbers of users
- Nick: 100Kcores is what DAsk is shooting for and it's purpose is to be torn up and down.
- Josh: slide 12 interoperability concerns, even C++ users will use python as a steering language
- Axel: python packages support interoperability for a wrapper strategy
- Jim: portability of python is superior , conda…
- Eduardo: there are functionality reasons (4-5D histograms) that people look for tools outside of the HEP ecosystem. The experiments need to think about how to properly support what the community wants regardless of toolset.
- Axel: is scarred … not the goal to separate the community, what happened to the contributions back to ROOT?
- Nick: Don't reinvent the wheel if you don't need to...

- Eduardo says he is not suggesting we get rid of ROOT… difference between what is optimal and what is actually done…
- Jim: want to be clear, not looking to replace ROOT, connecting everything to everything else… reduce barriers
- Axel: does the python analysis include pyroot? A: no
  Eduardo: dunno who said “no” but from personal knowledge it’s “yes” for some, “no” for others.
- Lindsey: separate Skema from underlying file format Josh: at some point we will have to pick one, Lindsey: it is already easy to convert from one to the other
- Graeme: archival format has different needs/ advantages then the format that runs best on your HPC

ROOT

- Axel Naumann presents for ROOT
- Paolo: your focus on speed but what about data compression? AL the team is constantly looking at compression even lossy compression (like what serves nanoAOD?) can these strategies be generalized? It’s R&D… requires experiment cooperation for validation.. Axel does not thing that validation can be done in general ; auto-differentiation could be a key tool for this topic
    - Christian Tacke: Doing the loss uses some "model". And the key question is, whether that model influences your models further down the road. So this is a really complex question.
- Doug: Analysis on HPC? Why? Axel: opportunistic use of them is still interesting, we should be able to understand it and that is why we are doing R&D there
- Nick: the issue is proliferation ; what we need is a framework that allows optimization more dynamically A: agrees RNTuple’s goal is to add the dataset abstraction, just moving away from files may not be the answer, the current thought is that experiments do the work to optimize skimming
- Jim: didn’t mean to give the wrong impression about the landscape change, he would characterize it as a slow change 3 changes in 25 years is slow. There maybe more like one or two more, but they don’t capsize the boat. It’s not completely unpredictable. Axel: agrees but it does seem that analysis evolves faster then the other pieces of our workflow… ML was first seen in analysis, we don’t know what it will look like in 10-15 years, what data formats will it need? Hopefully that need will not be orthogonal to today's trends Jim: we can say structures of arrays will still be useful, we will be using ML

AOB

- Axel: Understands columnar analysis as the data layout but there is also how you write the analysis? Do you still think about operations on event objects? A: Maria - had the event loop change in mind; Pete: we are not really always event based, we ask questions like “ give me all of the events with characteristic X” Lindsey: thinks about it as a merger of the data organization with verbs that let you operate on the

data in a specific way (even depending on the hardware). They ways of thinking about this are a false dichotomy ; there needs to be design engineering in addition to software engineering … you start with event thoughts and translate that to operations on sets of events

- Pete: its not just the dataformats but the language that is used to describe the analysis
- Lindsey: How do we talk about design in a more non-engineering way
- Lukas: it depends on the questions you want to ask and what stage of the analysis you are at. Thinks you need both types of abstractions TTreedraw is the old fashioned tool for this.
- Josh: Speed motivates lots of innovative changes Lindsey; thinking about event at a time vs. array at a time is at first jarring but then going back is easier; it would be worth thinking about how to make this easier Josh: if things were setup well you would not be able to tell the difference Pete: thisis a difficult problem and is why compilers don't auto-vectorize well.
- Jim: It's true that "columnar analysis" can mean the physical layout (can be hidden) and the method of interface (array-at-a-time "verbs"). These are conceptually distinct, but _having_ the efficient layout changes the performances of verbs in the array-at-a-time interface, making the latter something you're more likely to want to use if you have the former.
- Axel: +1 - indeed, and I wanted to understand which one people are referring to. I think I have my answer :-)
- Lindsey: event vs. columnar expressions probably largely comes down to how much you thrash the icache? or how much of a possibility there is to do so