UC SANTA CRUZ

CERN

Analysis design and facilities

Dr. Giordon Stark

HSFDAWG - ATLAS Analysis

March 23rd, 2020

indico.cern.ch/e/890991/

ATLAS EXPERIMENT

Run: 300800

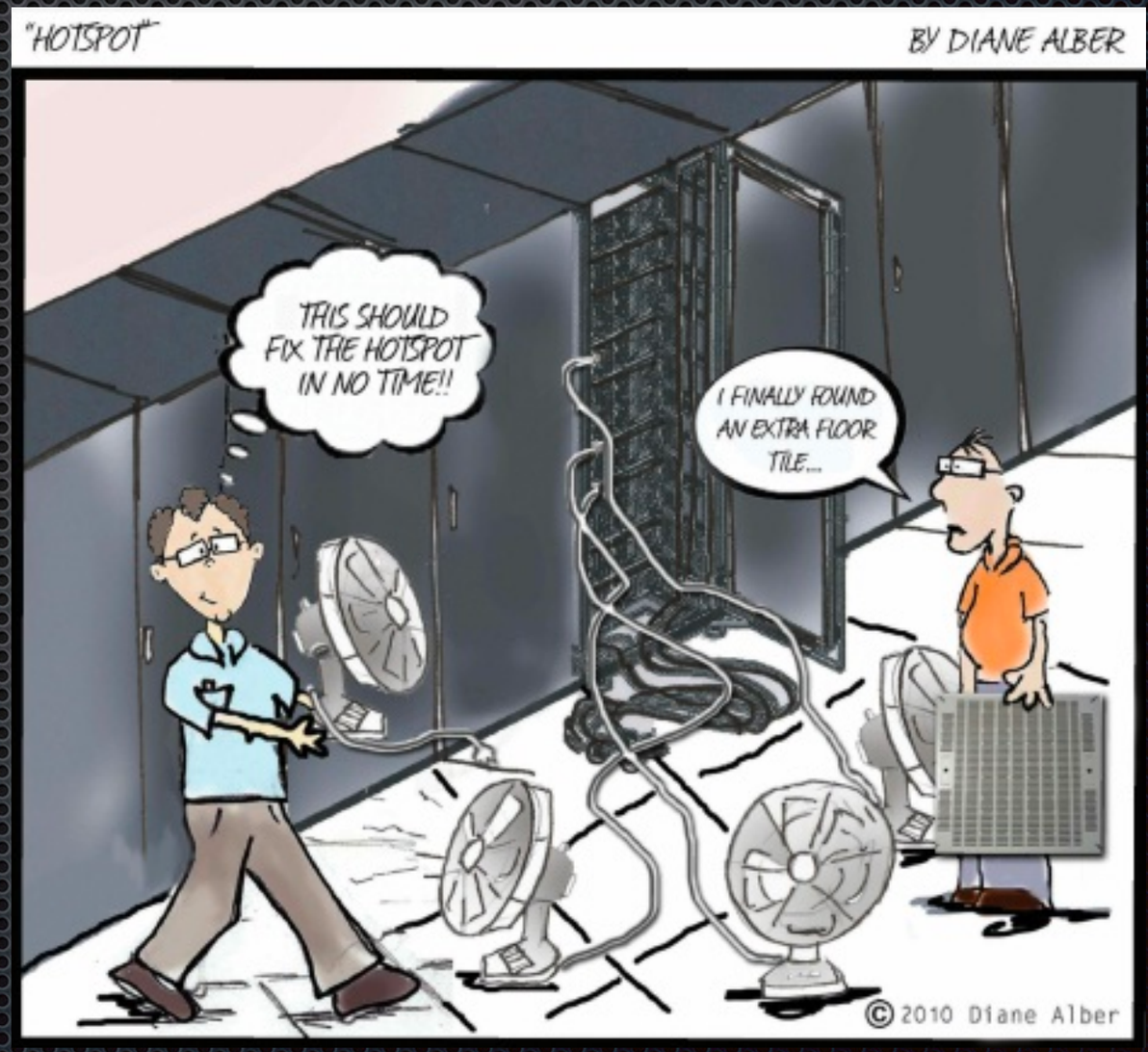
Event: 2418777995

2016-06-04 03:47:03 C

if you can read this, you're too close

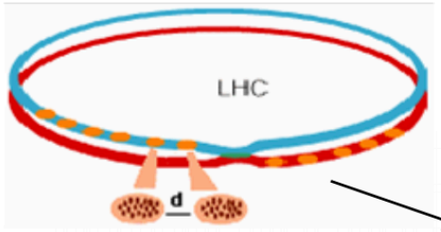
Goal

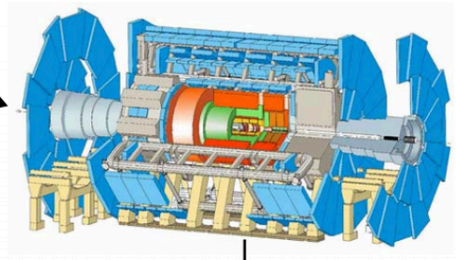
- Understanding ATLAS analyses
  - How should an analysis facility be tailored for a given ATLAS analysis
- ATLAS data access patterns
  - How should data be staged or cached?



ATLAS analyses

Steps from Data Collection to Physics Results





Data

Simulation

GENERATION — *Event generator output (EVNT)*

SIMULATION — *Simulated interaction with detector (HITS)*

Raw data (RAW) — TRIGGER

DIGITIZATION — *Simulated detector output (RDO)*

Calibration

Analysis object data (AOD) — RECONSTRUCTION

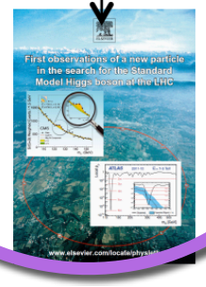
RECONSTRUCTION — *Analysis object data (AOD)*

Derived AOD (DAOD) — DERIVATION

DERIVATION — *Derived AOD (DAOD)*

ANALYSIS

Case Study



ATLAS-CONF-2018-041

Case Study: SUSY Multi-*b*-jets

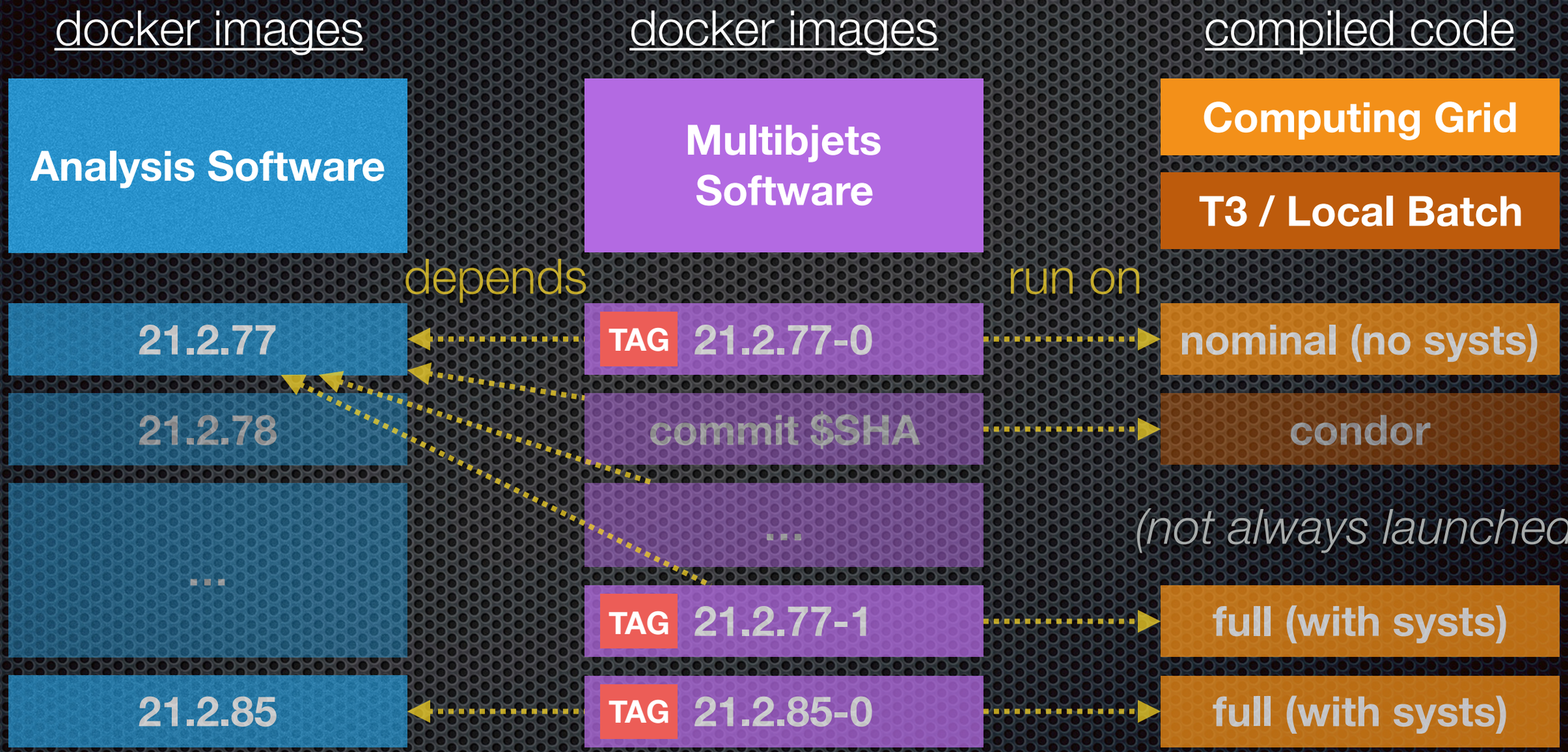
- Three data campaigns: mc16a, mc16d, mc16e
  - Analyse **254TB** across **1,730 DxAODs (SUSY10)** containing **5,408,224,942 events** in **156,930 files**. (Note: ~50kb/event)
  - End with **44 files** taking up **485 GB** of disk space with **85,522 flat ntuples** (trees; including systematics)
    NB: unoptimized ntuples have 1 systematic tree per "histogram", many of which are ~duplicates (very large overlap/duplication)
- What best practices do we do?
  - ✔ Implement continuous Integration
  - ✘ No checks for code style/code quality/coverage (very hard to do with current offline analysis software)
  - ✔ Automate docker image building of analysis for RECAST/REANA
  - ✔ Regression testing of physics
  - ✔ Git + Merge Request workflow for new features/changes

See IRIS-HEP for other case studies

Other analyses will have different tick marks

ATLAS-CONF-2018-041

Case Study: SUSY Multi-*b*-jets



Software: asynchronous updates

ATLAS-CONF-2018-041

Case Study: SUSY Multi-*b*-jets

⚠ Current struggles with launching full productions

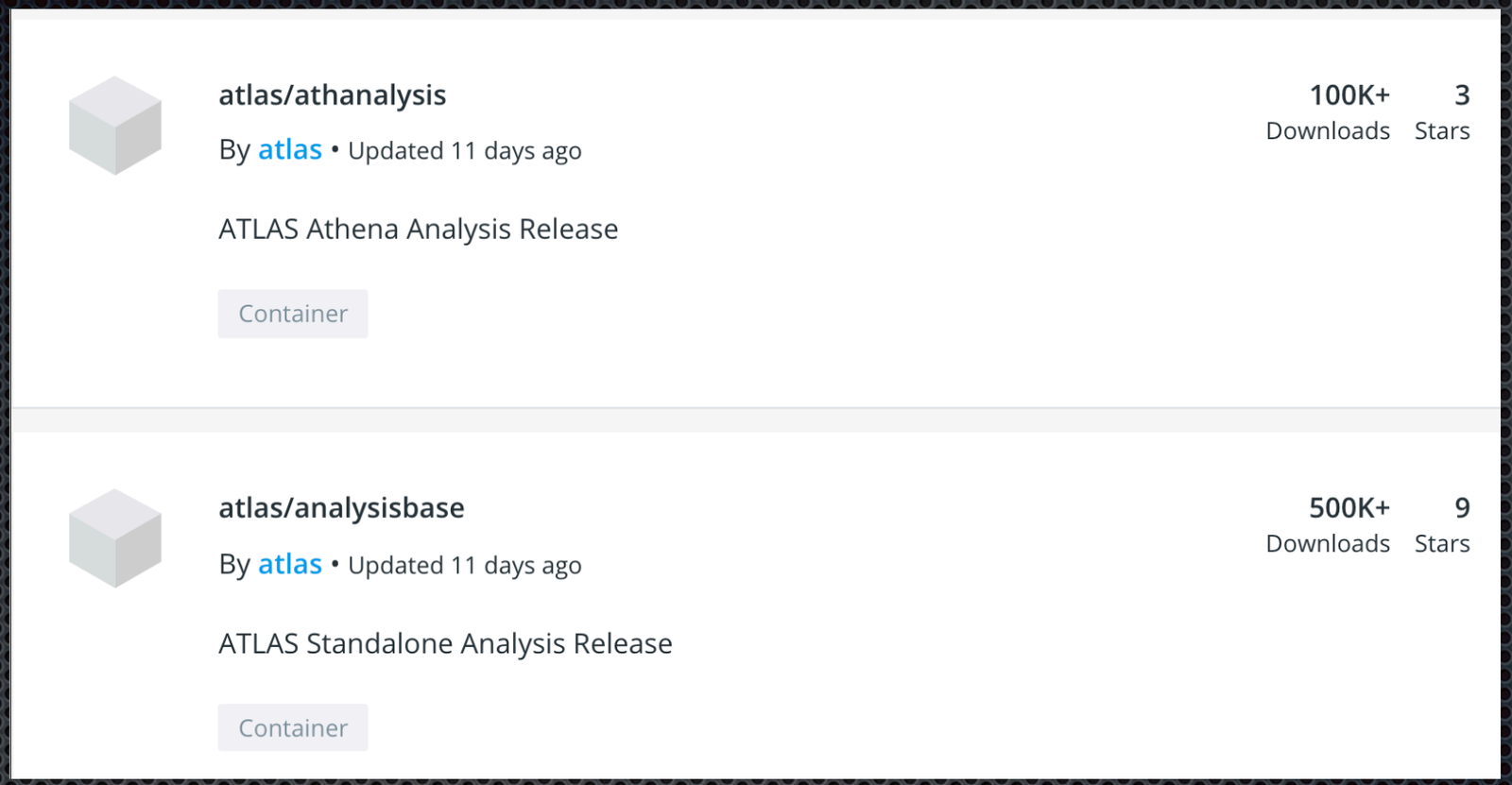
need to launch these 9-12 times per year

- 🐞 constant retries and resubmissions due to non-user failures
  - site issues (including: staging, caching, pilot)
  - sporadic bugs
  - lack of software updates or CVMFS sync
  - etc…
- Must distribute submission across multiple users 👥
  - a single user does not have enough quota for a single Multi-*b*-jet production
- Lack of containerization 🐳 on computing resources
  - Submission is slow as one must upload compiled binaries/code: O(500MB)
  - **Not enough sites support using containers (⚠ my opinion: biggest problem)**

Nominally: expect up to **1 week** to run over everything

Practically: takes up to **1 month** (including retries/resubs)

Quick aside: containerization



- Docker images with entirety of ATLAS offline software already have surpassed 1 million downloads

We need improved support for docker images

AMG: changes to datasets

- Instead of many samples made in N≫1 derivations covering many analyses, moving to a Run-III model with two new centralized derivations:
  - DAOD_PHYS (~50kb/event); storage: ~20PB/year
  - DAOD_PHYSLITE (~10 kb/event); storage: ~4PB/year

(see backup for breakdown)

- ✓ **Benefits:**
  - greatly reduce disk space for ~80% of analyses *(nb: educated guess!)*
  - improve dynamic caching (less variety of datasets to be cached, increases likelihood of reused datasets) — should reduce some site errors due to data staging
  - columnar data access: stream/on-disk via ROOT, Parquet, or other HEP analysis tools
- ✗ **Cons:**
  - More complex/unique analyses that cannot use DAOD_PHYS* continue to rely on custom derivations (Run-II model)
  - Exploring mitigation strategies such as “friend DAODs” or augmentation of PHYSLITE

⚠ PHYSLITE is Run 4 target format

AMG: changes to datasets

- Instead of many samples made in N≫1 derivations covering many analyses, moving to a Run-III model with two new centralized derivations:
  - DAOD_PHYS (~50kb/event); storage: ~20PB/year
  - DAOD_PHYSLITE (~10 kb/event); storage: ~4PB/year

(see backup for breakdown)

- ✔ **Benefits:**
  - greatly reduce disk space for ~80% of analyses *(nb: educated guess!)*
  - improve dynamic caching (less variety of datasets to be cached, increases likelihood of reused datasets) — should reduce some site errors due to data staging
  - columnar data access: stream/on-disk via ROOT (maybe Parquet, or other HEP analysis tools?)
- ✔ **Additional Benefits**
  - Case study Multi-*b*-jets analysis can reduce inputs by a factor of 5 — faster turn-around time on computing resources

❓ Computing Facilities: Do we have capabilities to regenerate datasets on demand?

ATL-SOFT-PROC-2020-002

AMG: changes to datasets

Table 3: Overview of a simple disk space model with Run-2 numbers and the AMSG-R3 recommendations applied as described in the text.

	MC				Data			
	AOD	DAOD	DAOD PHYS	DAOD PHYS LITE	AOD	DAOD	DAOD PHYS	DAOD PHYS LITE
events	$3 \cdot 10^{10}$	$1 \cdot 10^{11}$	$3 \cdot 10^{10}$	$3 \cdot 10^{10}$	$2 \cdot 10^{10}$	$1 \cdot 10^{11}$	$2 \cdot 10^{10}$	$2 \cdot 10^{10}$
size/event [kB]	600	100	70	10	400	50	40	10
disk space [PB]	18.0	10.0	2.1	0.3	8.0	5.0	0.8	0.2
other versions	1.5	2	2	2	1.5	2	2	2
repl. fac.	0.5	1	4	4	0.5	2	4	4
Sum [PB]	13.5	20.0	16.8	2.4	6.0	20.0	6.4	1.6

Already 50% smaller than a typical DAOD!

⚠ **Looking at 85PB/year - but potentially could free up 46PB/year of that (allows for more statistics when making MC)**

AMG: changes to datasets

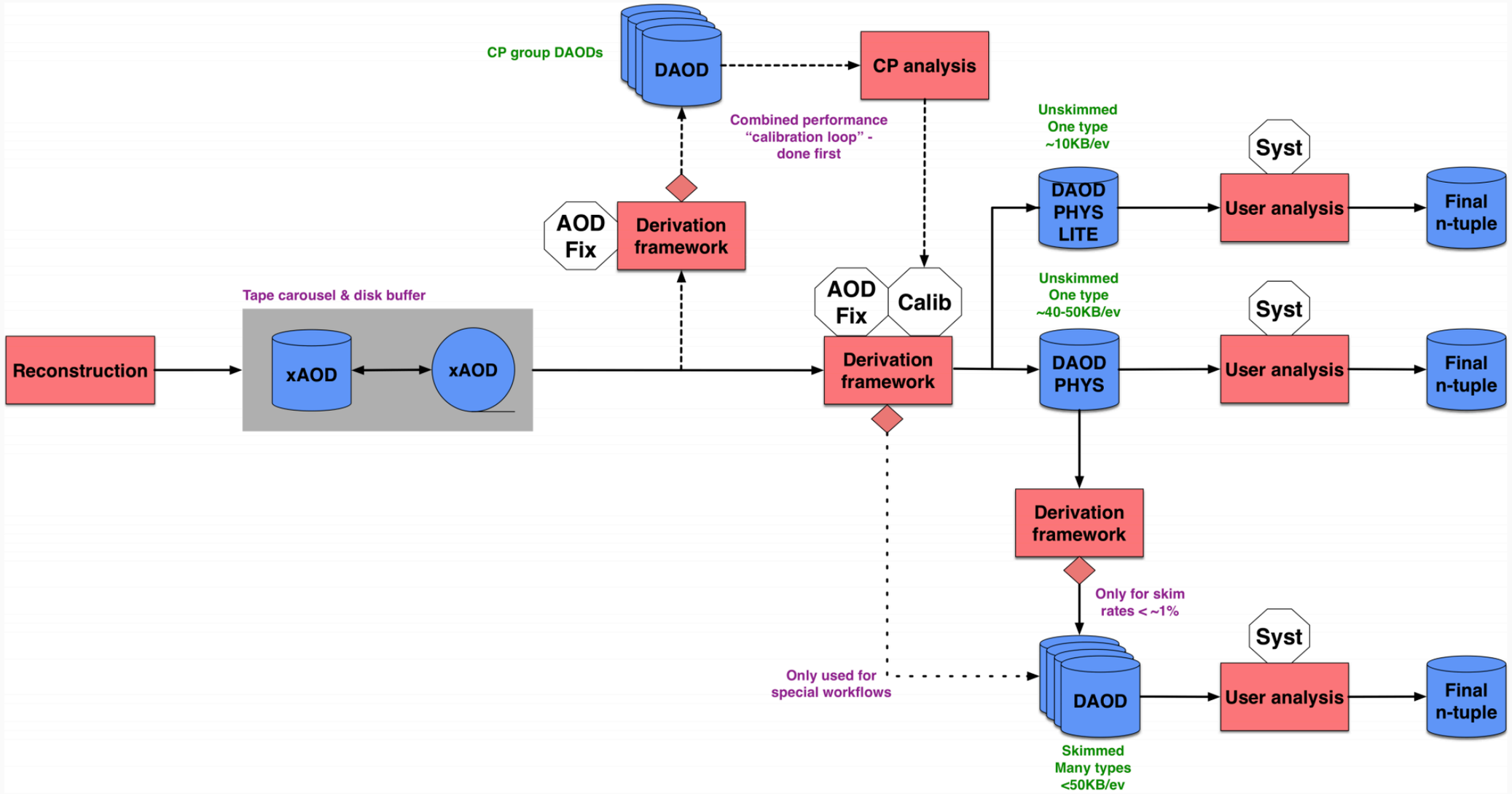
Caveats

- Using an analysis framework that produces ntuples
- Many CP tools are setup and executed (even if not needed - PHYSLITE does not need)
- Run on local machine
- No systematics

	PHYSLITE [events/s]	SUSY5 [events/s]
Run 1	117.43	105.94
Run 2	123.69	106.78
Run 3	118.93	108.99
Run 4	120.86	106.34
Run 5	117.85	109.16
Run 6	121.90	104.81
Average	120.11	107.00

Up to a 20% speed up!

AMSG-R3: New production workflows and formats



DAOD_PHYS:
50 kB/event, combined single DAOD format (for MC, but also DATA), AOD event data model (EDM)

DAOD_PHYSLITE:
10 kB/event, very condensed and calibrated objects, very important for HL-LHC, AOD or ntuple EDM, ideal for DOMA/XCache

today's DAODs:
Significantly reduce number of today's DAODs

AODs:
Larger fraction only available on TAPE

AMG: changes to analysis

- Multi-*b*-jets has the following transformation graph

- Take advantage of new tools
  - (ROOT7) RDataFrame, RNTuple
  - (PyHEP) numpy, scipy, uproot, coffea, ServiceX, pyhf, others (🔨🪛🔧)?
- and new ideas
  - Multi-threading?
  - Co-processing (GPUs, FPGAs)?
  - Data Lakes?
  - More containerization?
  - Machine Learning?

❓ PHYSLITE could be small enough (pre-filters) to go directly to histograms?

(no need for intermediary format, or extra processing step!)

AMG: changes to analysis

In [22]:
```
import uproot
import awkward
import uproot_methods
import matplotlib.pyplot as plt
%matplolib inline
```

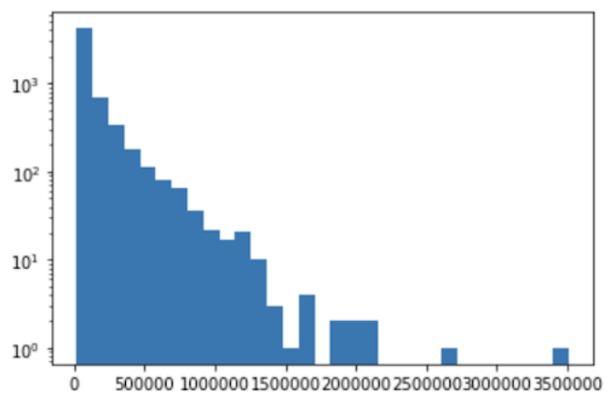
```
UsageError: Line magic function `%matplolib` not found.
```

In [23]:
```
f = uproot.open('./valid1.410000.PowhegPythiaEvtGen_P2012_ttbar_hdamp172p5_nonallhad.DAOD_PHYSVAL.e4993_s3189_r9524_At
t = f['CollectionTree']
```

In [24]:
```
vectors = uproot_methods.TLorentzVectorArray.from_ptetaphim(
    t['AntiKt4EMTopoJetsAux.pt'].array(),
    t['AntiKt4EMTopoJetsAux.eta'].array(),
    t['AntiKt4EMTopoJetsAux.phi'].array(),
    t['AntiKt4EMTopoJetsAux.m'].array(),
)
```

In [29]:
```
plt.hist(vectors.E.content,bins = 31)
plt.semilogy()
```

Out[29]:
```
[]
```



caveats: element-links?

In [ ]:

PHYSLITE read via “native” columnar (e.g. uproot)

Wishlist

❗ infrastructure that allows rapid analysis on HL-LHC scale data volumes

- Prefer not to centralize resources in large-scale computing facilities
  - Hands-on/manage the resources during peak usage, but ensure that the hardware is not unused the rest of the time
- Dynamic capability of enabling high IOPS jobs / GPU-required jobs
  - e.g. on-demand mounting/deploying of high performant storage
- Interactive, user-friendly (student and “non-HEP-only" friendly) technology such as jupyter
  - Need to ensure equitable distribution of these resources (“fair share”)
- Want to be able to R&D / trial various solutions to find one that best meets our needs
  - for example: jupyter hub + dask/spark backend scheduled dynamically over k8s clusters
  - once a solution is chosen, reconsolidate brain power to focus development

❓ Is there a federated approach in which facilities / interactive analysis modes could be integrated into ATLAS resources?

Backup

Table 3: Overview of a simple disk space model with Run-2 numbers and the AMSG-R3 recommendations applied as described in the text.

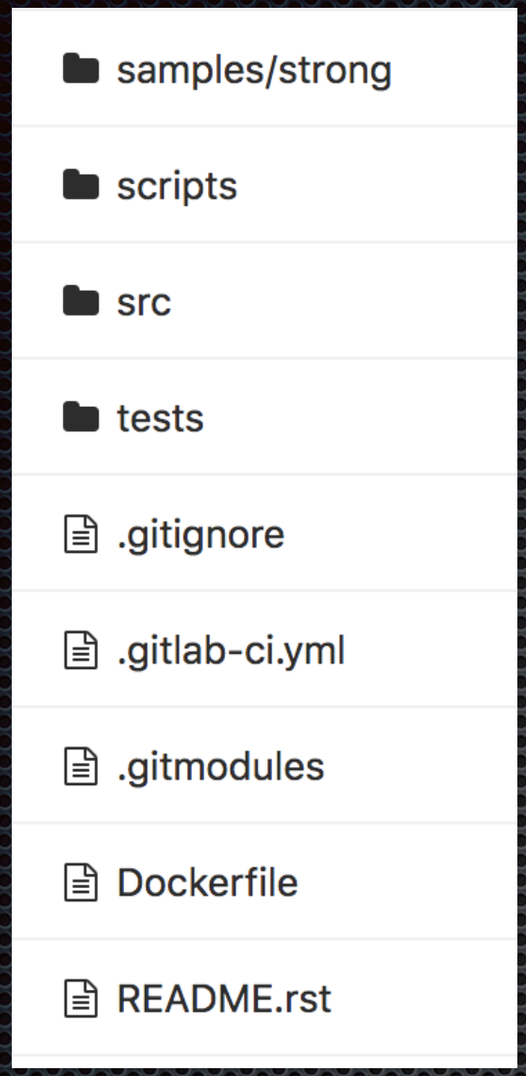
	MC				Data			
	AOD	DAOD	DAOD PHYS	DAOD PHYS LITE	AOD	DAOD	DAOD PHYS	DAOD PHYS LITE
events	$3 \cdot 10^{10}$	$1 \cdot 10^{11}$	$3 \cdot 10^{10}$	$3 \cdot 10^{10}$	$2 \cdot 10^{10}$	$1 \cdot 10^{11}$	$2 \cdot 10^{10}$	$2 \cdot 10^{10}$
size/event [kB]	600	100	70	10	400	50	40	10
disk space [PB]	18.0	10.0	2.1	0.3	8.0	5.0	0.8	0.2
other versions	1.5	2	2	2	1.5	2	2	2
repl. fac.	0.5	1	4	4	0.5	2	4	4
Sum [PB]	13.5	20.0	16.8	2.4	6.0	20.0	6.4	1.6

Table 4: Overview of a very simple disk space model extrapolation for the HL-LHC applying the AMSG-R3 recommendations as described in the text.

	MC			Data			Sum
	AOD	DAOD	DAOD PHYSLITE	AOD	DAOD	DAOD PHYSLITE	
events (25-28)	$6.4 \cdot 10^{11}$			$1.5 \cdot 10^{11}$			
events / year	$2.1 \cdot 10^{11}$	$1.1 \cdot 10^{12}$	$2.1 \cdot 10^{11}$	$5.0 \cdot 10^{10}$	$2.5 \cdot 10^{11}$	$5.0 \cdot 10^{10}$	
size/event [kB]	1000	100	10	700	50	10	
disk [PB/year]	213.3	106.7	2.1	35.0	12.5	0.5	369.6

gitlab://MultiBJets/MBJ_Analysis/

Software Layout - MBJ (I)



Top-level contains source code, tests, and miscellaneous (short) scripts

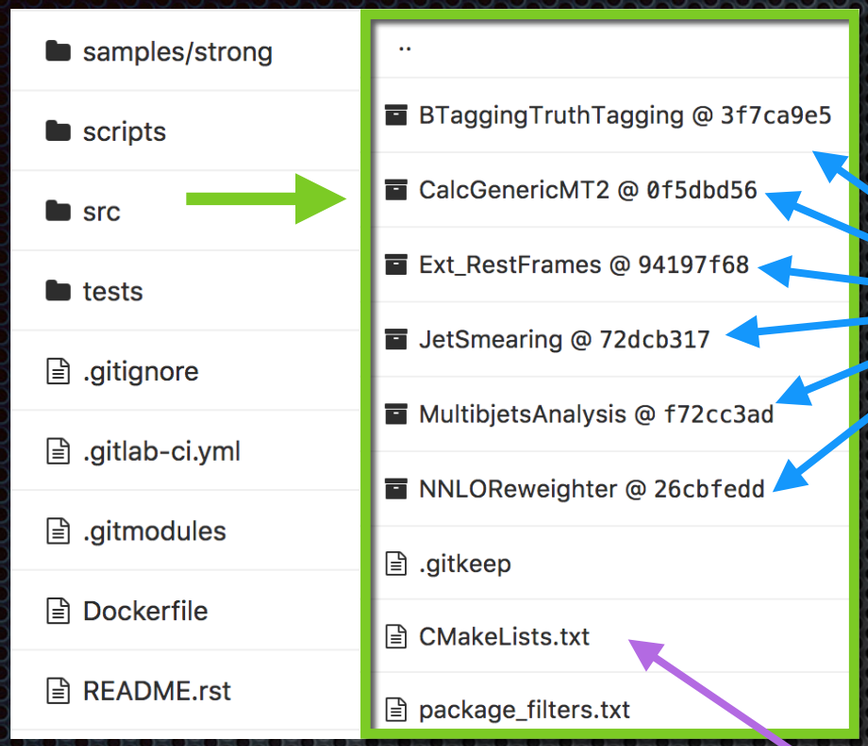
- source code uses a **submodule layout** (see next slide, `.gitmodules`)
- Add a README so users know whats up
- Take good care of your gitignore, many lazy users run `git add .` — adding all files — causes repository bloat and annoyed physicists
- `.gitlab-ci.yml` defines continuous integration (see later slides)
- `Dockerfile` — instructions on preserving code and runtime environment

Dockerfile

```
FROM atlas/analysisbase:21.2.67
COPY mbj_analysis.rpm /code/mbj_analysis.rpm
RUN sudo rpm -i /code/mbj_analysis.rpm && \
    sudo rm -rf /code/mbj_analysis.rpm && \
    sudo chmod 666 /home/atlas/release_setup.sh && \
    sudo printf '\n# Set up the MBJ code\nsource /usr/MBJ_Analysis/${AtlasVersion}/InstallArea/${AnalysisBase_PLATFORM}/
setup.sh\necho "Configured MBJ from: $MBJ_Analysis_DIR"' >> /home/atlas/release_setup.sh && \
    sudo chmod 644 /home/atlas/release_setup.sh
```

gitlab://MultiBJets/MBJ_Analysis/

Software Layout - MBJ (II)



`.gitmodules`

```
[submodule "src/MultibjetsAnalysis"]
        path = src/MultibjetsAnalysis
        url = ../MultibjetsAnalysis
[submodule "src/BTaggingTruthTagging"]
        path = src/BTaggingTruthTagging
        url = ../BTaggingTruthTagging
[submodule "src/Ext_RestFrames"]
        path = src/Ext_RestFrames
        url = https://github.com/lawrenceleejr/Ext_RestFrames
[submodule "src/JetSmearing"]
        path = src/JetSmearing
        url = ../../atlas-phys-susy-wg/JetSmearing
[submodule "src/NNLOReweighter"]
        path = src/NNLOReweighter
        url = ../NNLOReweighter
[submodule "src/CalcGenericMT2"]
        path = src/CalcGenericMT2
        url = ../../atlas-phys-susy-wg/CalcGenericMT2
```

Reduce the number of `git clone` a user calls manually

- **submodule layout** is a one-stop shop for your entire code
  - "**symbolic link**" to other packages and source code at a specific revision
  - grab everything with a single (recursive) git clone
    - Contains a top-level `CMakeLists.txt` which you should write (ASG recommends writing your own!)
- Use the same layout for checking out code which reduces user error and standardizes the workflow
- Top level CMakeLists.txt is committed — as it should be written for each analysis code separately (ASG recommendation)
- Use relative submodule paths where possible, as this allows cloning over kerberos, ssh, or https automatically
  - Also integrates very nicely with continuous integration