Needs and Concerns

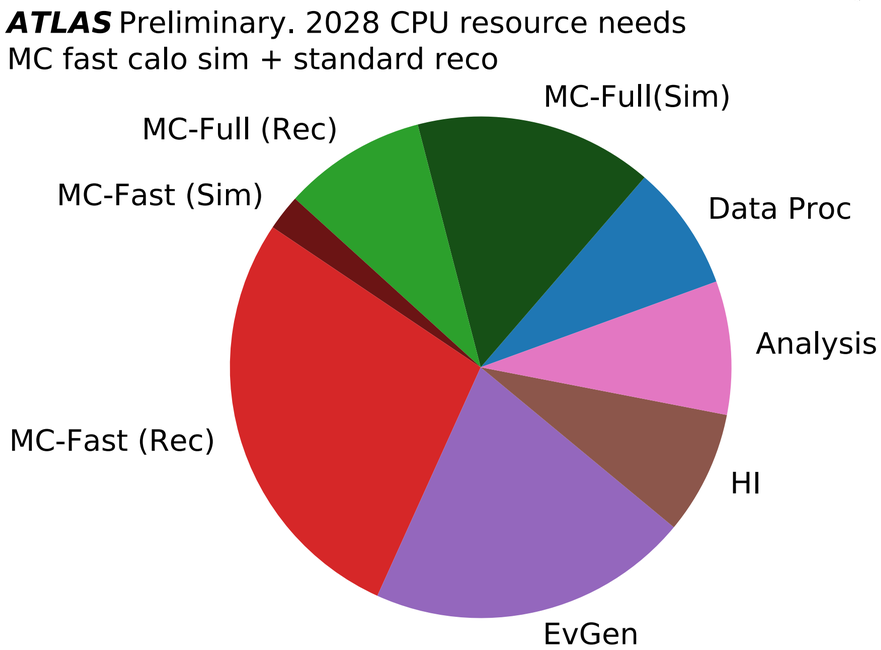
Event Generator Computing Workshop
26/11/2018

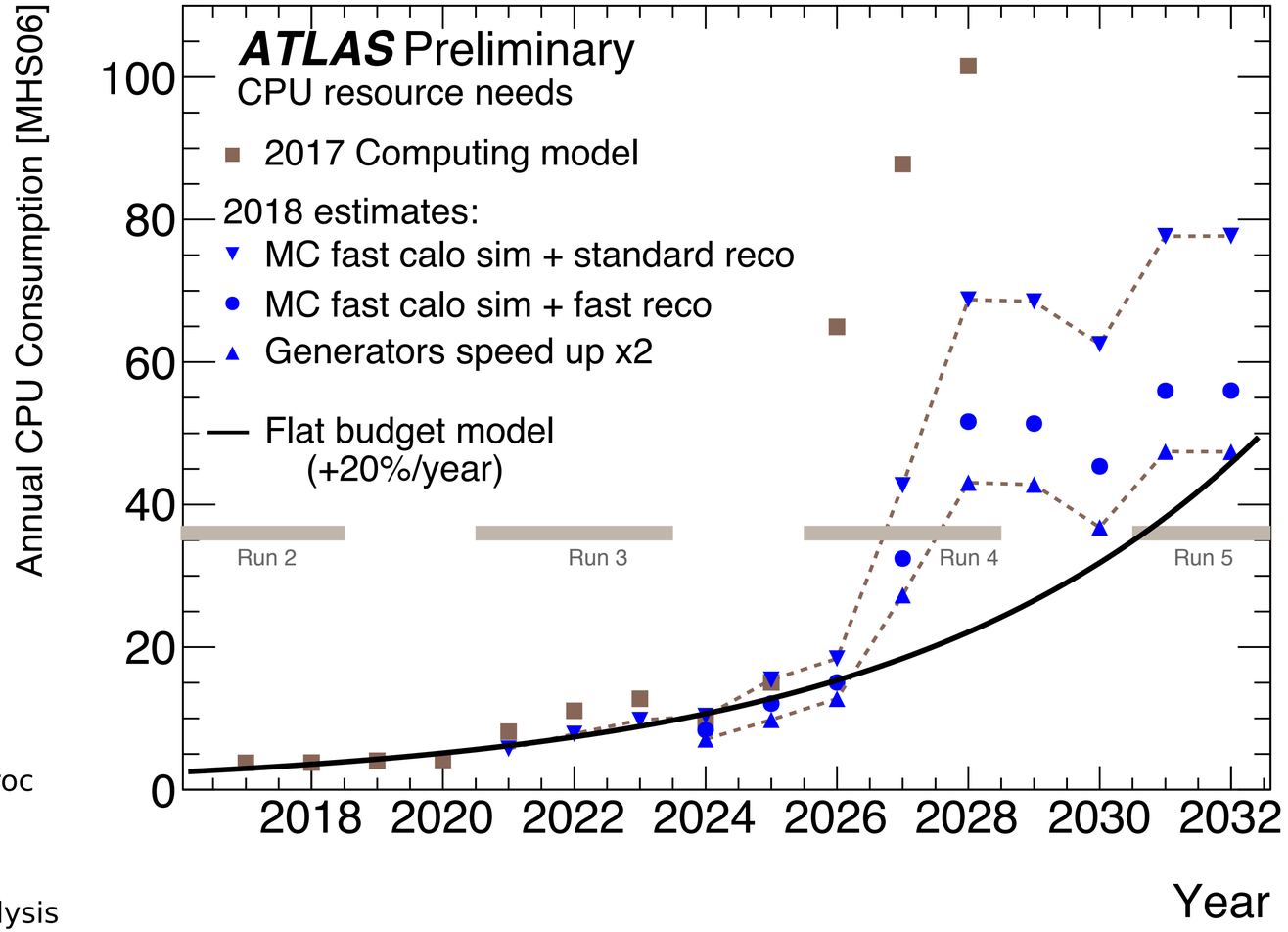
Josh McFayden,
Simone Amoroso & Frank Siegert
for ATLAS

ATLAS EXPERIMENT

Setting the scene | Motivation

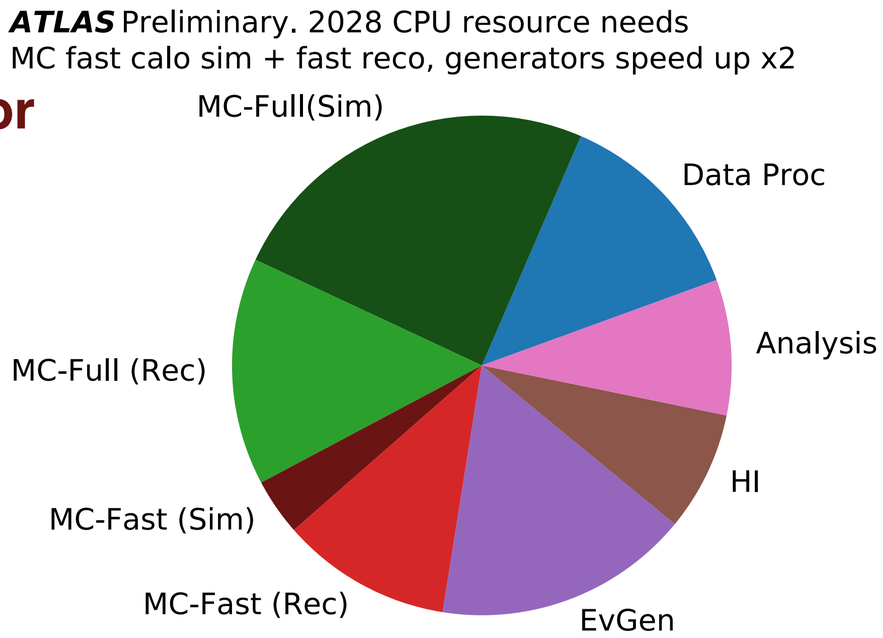
- The motivation for closer scrutiny of **MC generation** resource usage is clear:
  - The current model does not scale much beyond Run 3

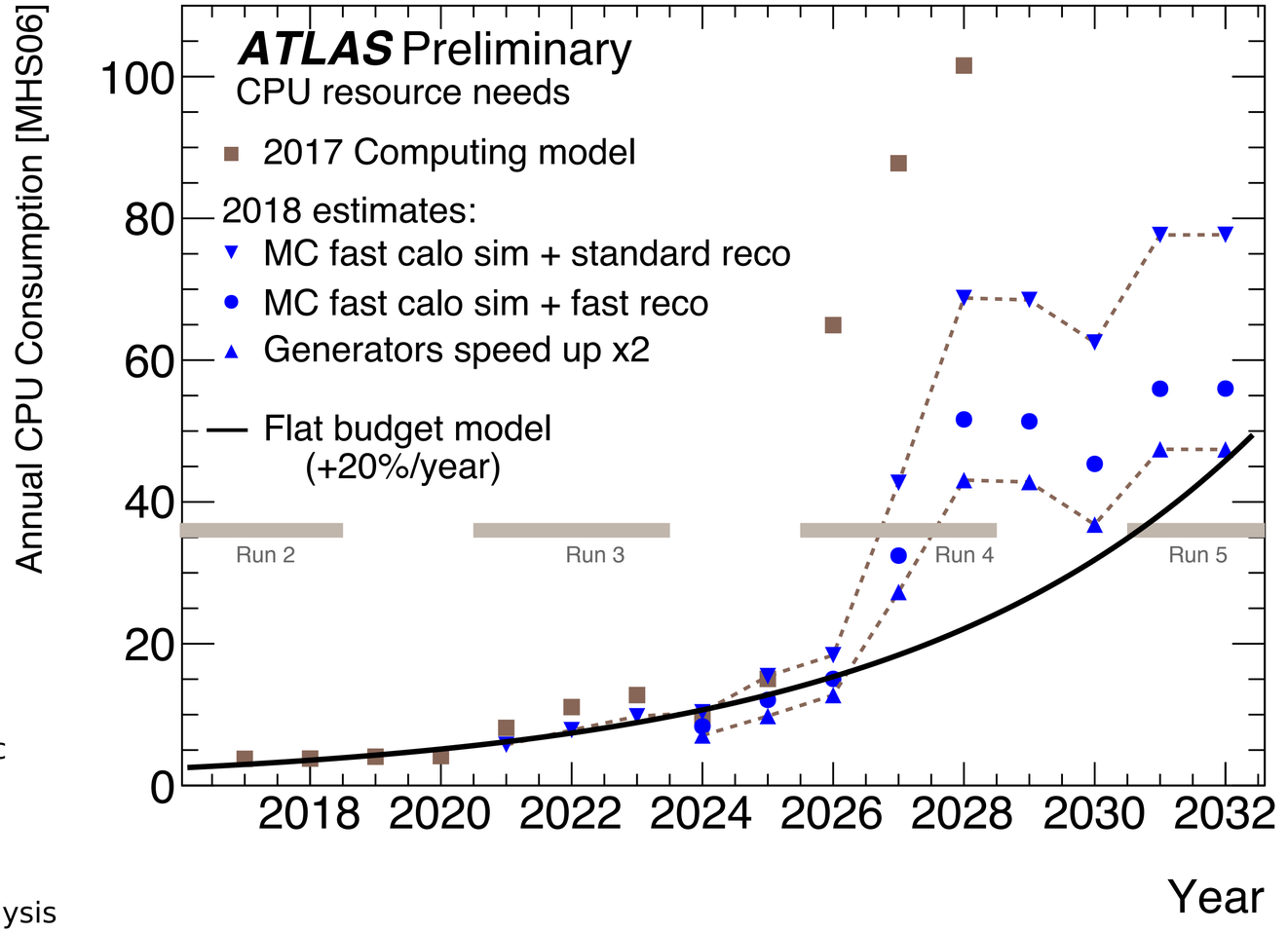




Setting the scene | Motivation

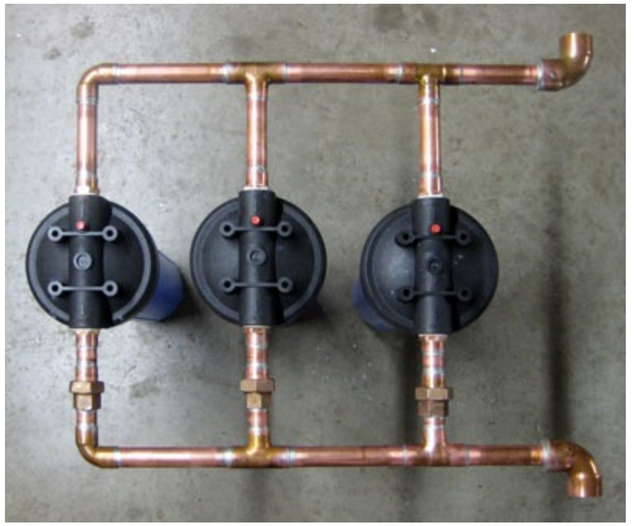
- The motivation for closer scrutiny of **MC generation** resource usage is clear:
  - The current model does not scale much beyond Run 3
  - The fraction of resources dedicated to event generation is about to dramatically increase
    - Because **fast detector simulation** will be used for a much larger fraction of events.



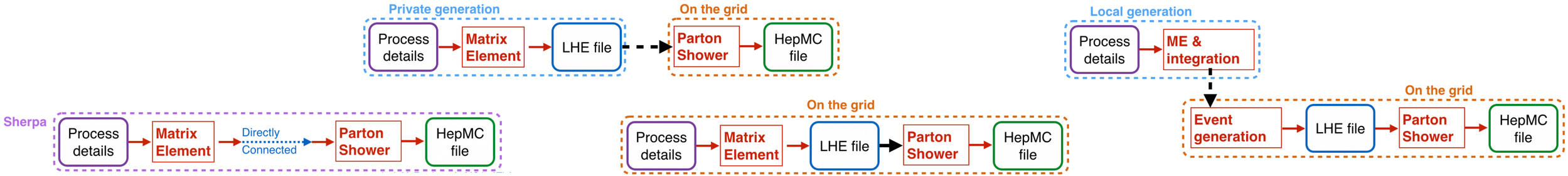


Generator implementation basics

- All generator code is external software for experiments.
- Interface packages are written to make the input and output Athena-friendly.
- Different generator codes have different…
  - physics processes
  - available precision
  - technical features
  - input/output needs

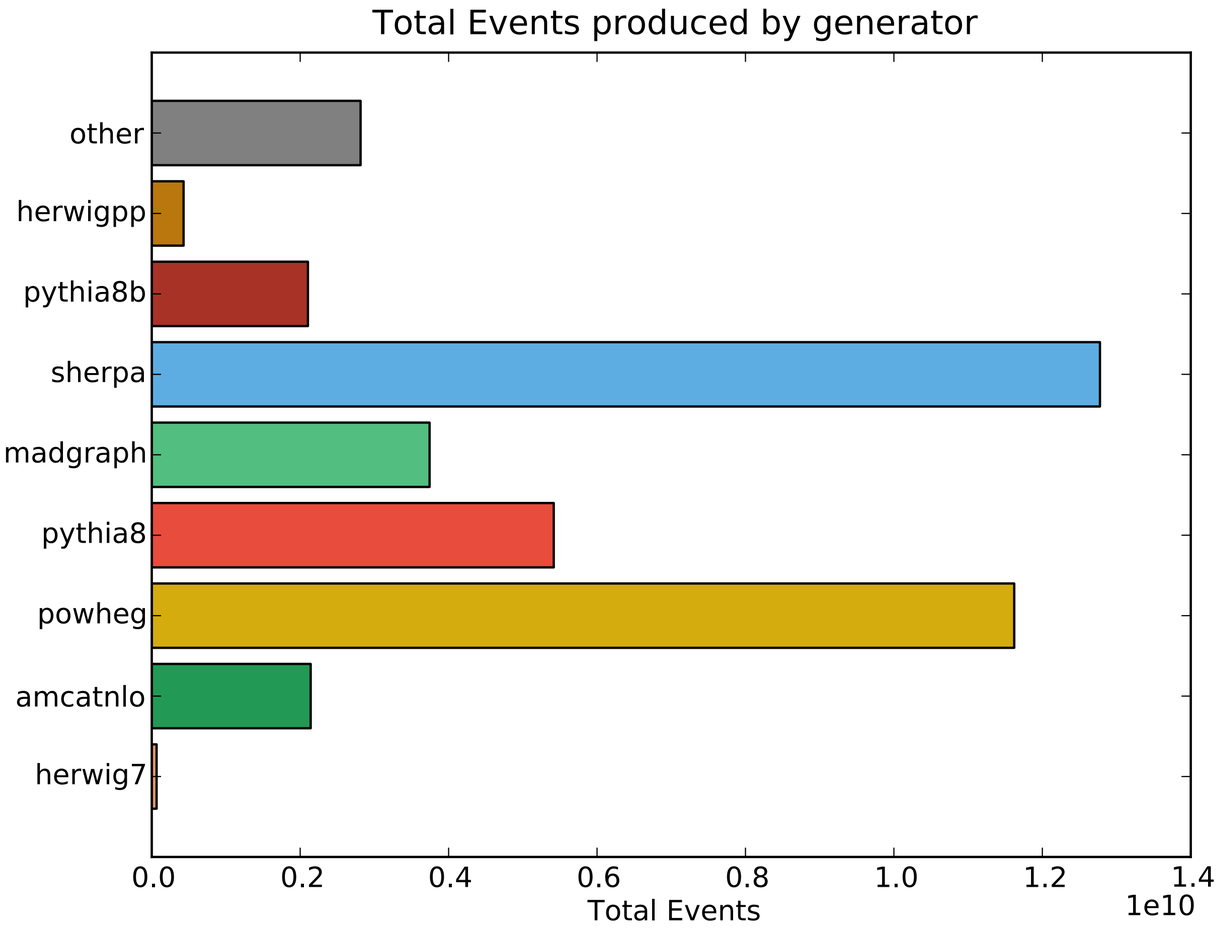


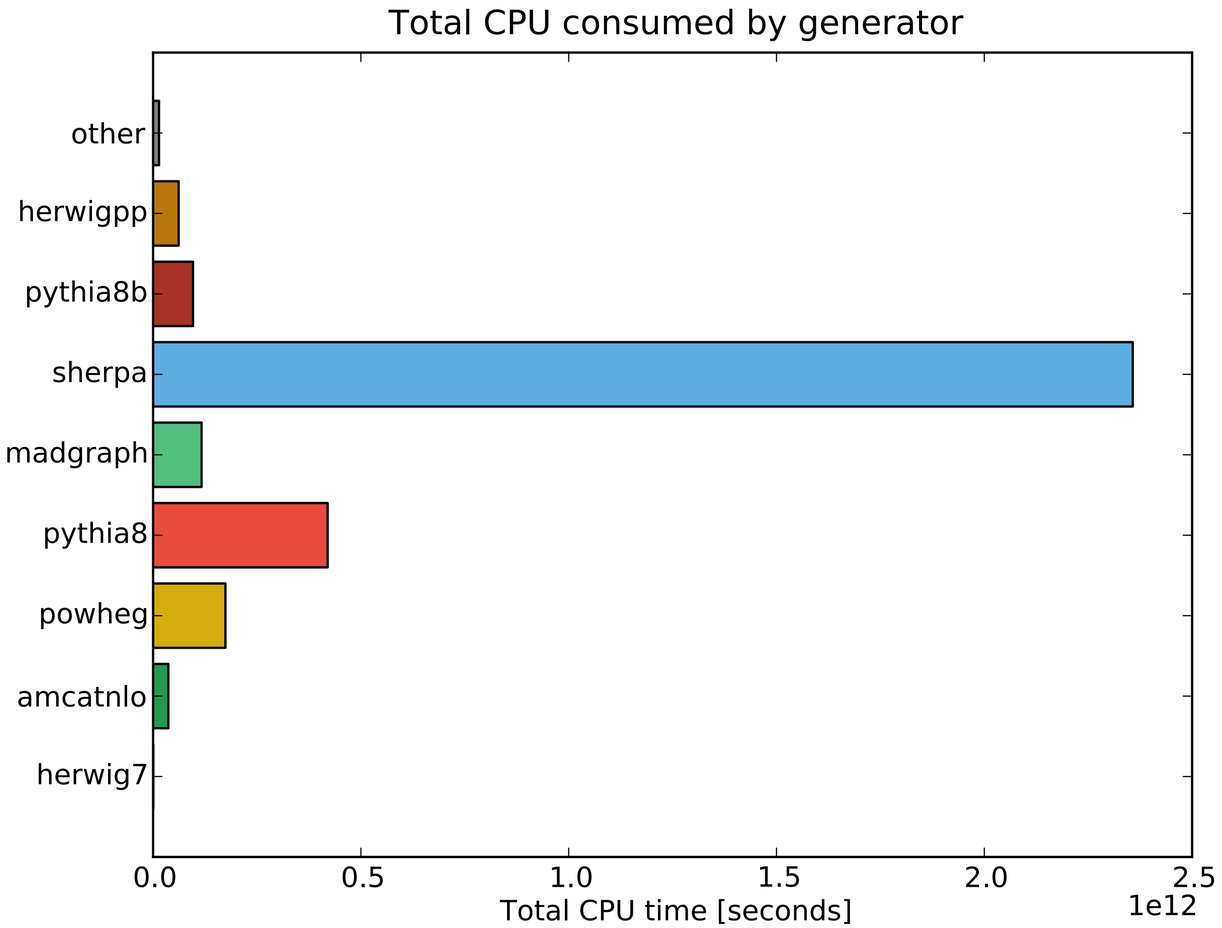
- Various possible configurations result in many different running modes
  - Also requires flexibility in the software integration and production system configuration.



Current generator usage | Total events and CPU

- Majority of CPU consumption comes from Sherpa2.2 V+jets setups.
  - But these are by far our largest (3.2B events) and most precise (V+0,1,2j@NLO+3,4j@LO) samples

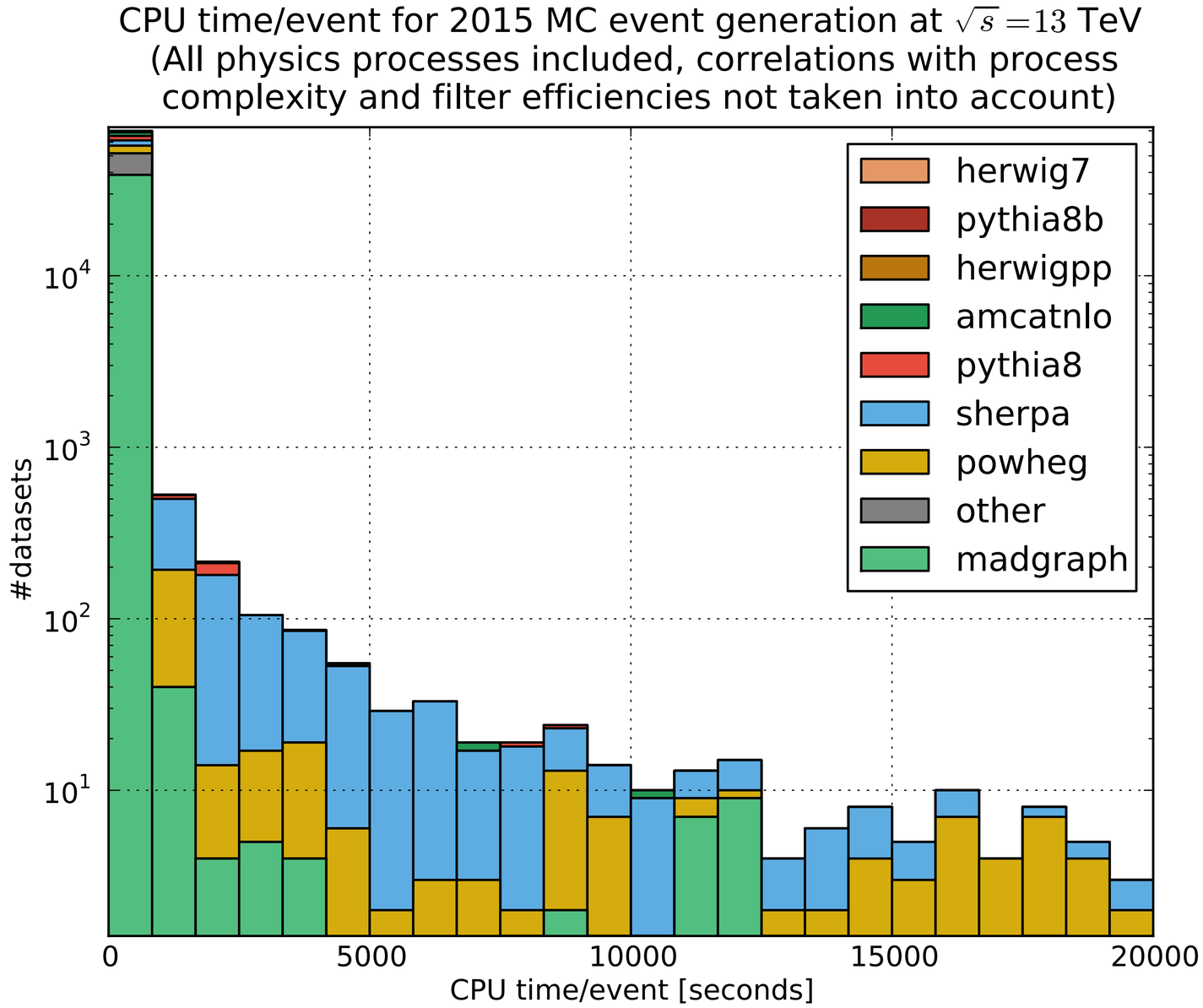




Current generator usage | Avg CPU/event

	Avg CPU/evt
Evgen	~80 s
FullSim	~245 s
FastSim	~45 s
Reco	~60 s

- Average CPU/event by dataset
  - Each entry is a single dataset
  - The numbers have not been corrected for e.g. filter efficiencies
  - ME generators also include showering
    - Sometimes only showering… but it's hard to separate
- The average across all samples is ~80s/event
- But have many examples where event generation is slower than full simulation…
  - A strong indication that this is something to improve!

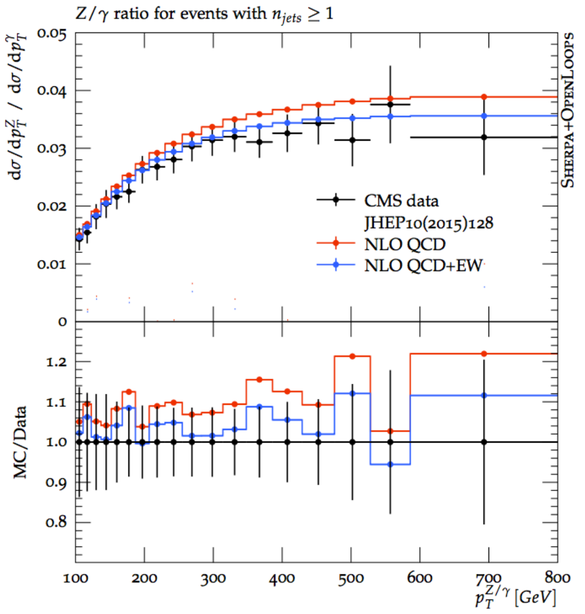


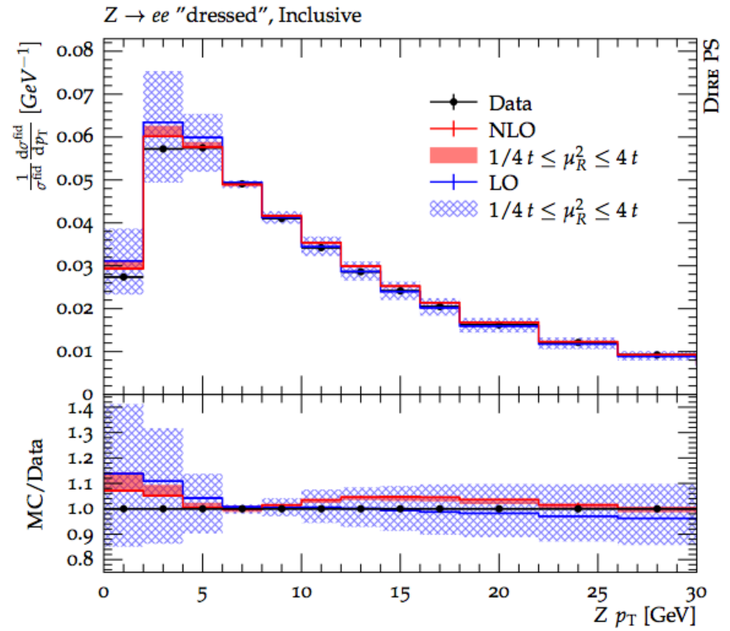
Recent improvements

- We have seen very impressive increases in generator precision over the last years
  - NLO merging
  - NLO EWK corrections
  - NNLO QCD corrections in some cases

- Systematic weights
  - The implementation of systematic ME scale, PDF and shower scale variations has significantly improved the efficiency of our event generation!

- **This work is of course greatly appreciated!**





Problems

- Negative weights
  - We cannot afford to run full simulation on samples with negative weight fraction >25%
    - Starting to become a deal-breaker for some setups
    - High statistics W/Z samples for precision analyses cannot currently use MC@NLO-like matching schemes.
- Precision vs CPU
  - The increase in precision goes hand-in-hand with increases CPU consumption.
  - If we want increased precision in future we will need even more CPU!
- Efficiently populating extreme regions of phase space
  - Filtering and generating events in extreme regions of phase space can be problematic…

ttbar production:

Sample	Fraction of events with neg. weights [%]
Sherpa (lepton+jets)	20.5
Sherpa (lepton+jets)	20.4
Sherpa (dilepton)	20.4
Sherpa ttbb (lepton+jets, CSSKIN, 4FS)	24.4
Sherpa ttbb (lepton+jets, CMMPS, 4FS)	25.7
aMC@NLO+Py8 (lepton+jets)	23.7
aMC@NLO+Py8 (dilepton)	23.7
aMC@NLO+Py8 (FxFx, 70 GeV)	28.4
aMC@NLO+H++ (4FS, ttbb)	37.2
Powheg+Herwig7 (lepton+jets)	0.4
Powheg+Herwig7 (dilepton)	0.4

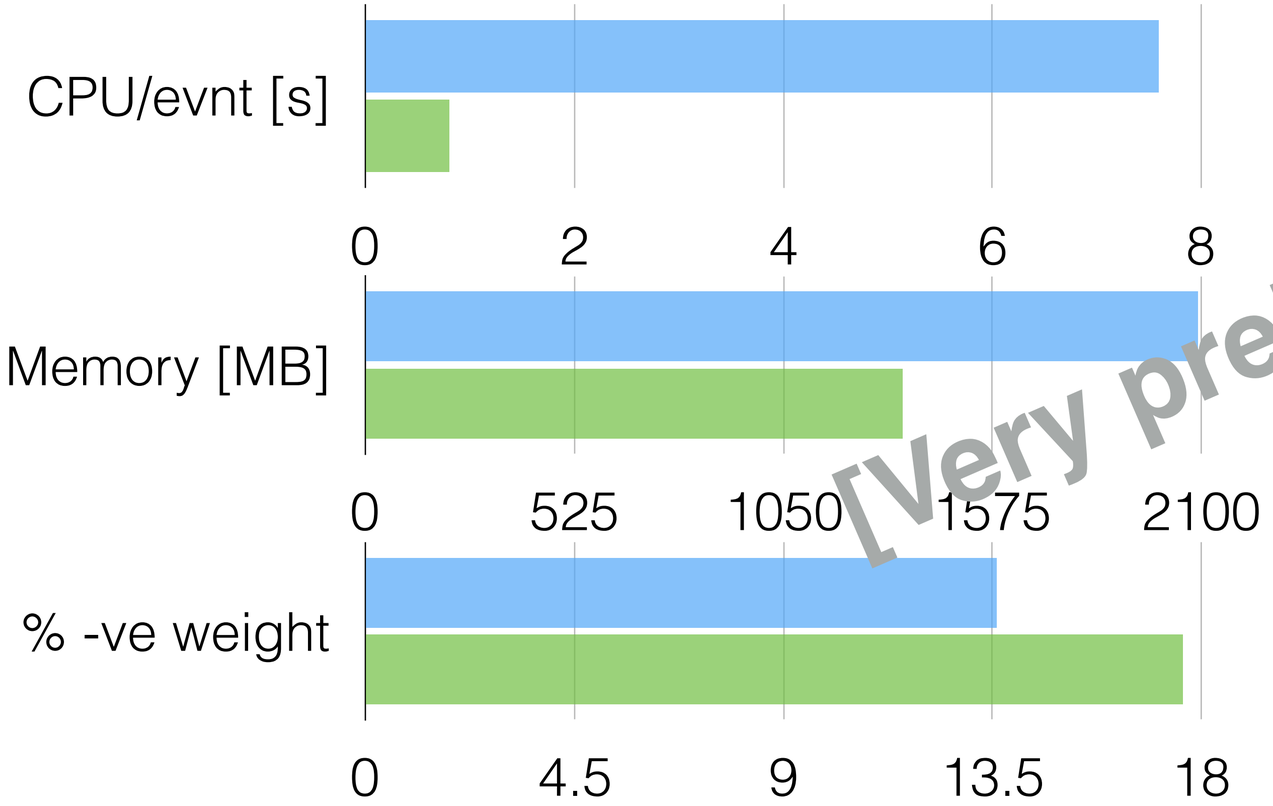
Possible routes out

- Work ongoing in the generator/tools groups to both increase the CPU efficiency and reduce negative weights
  - But this work is unglamorous and does not get you papers or permanent jobs…
  - Can experiments do more to help?
    - At the end of the summer ATLAS and CMS MC generators conveners asked for feedback on the **possiblity to fund positions in this area**.
    - Feedback was generally positive
      - There were quite strongly differing opinions on the implementation & shared fears on finding the right candidates.
      - Hopefully we can discuss more tomorrow morning…

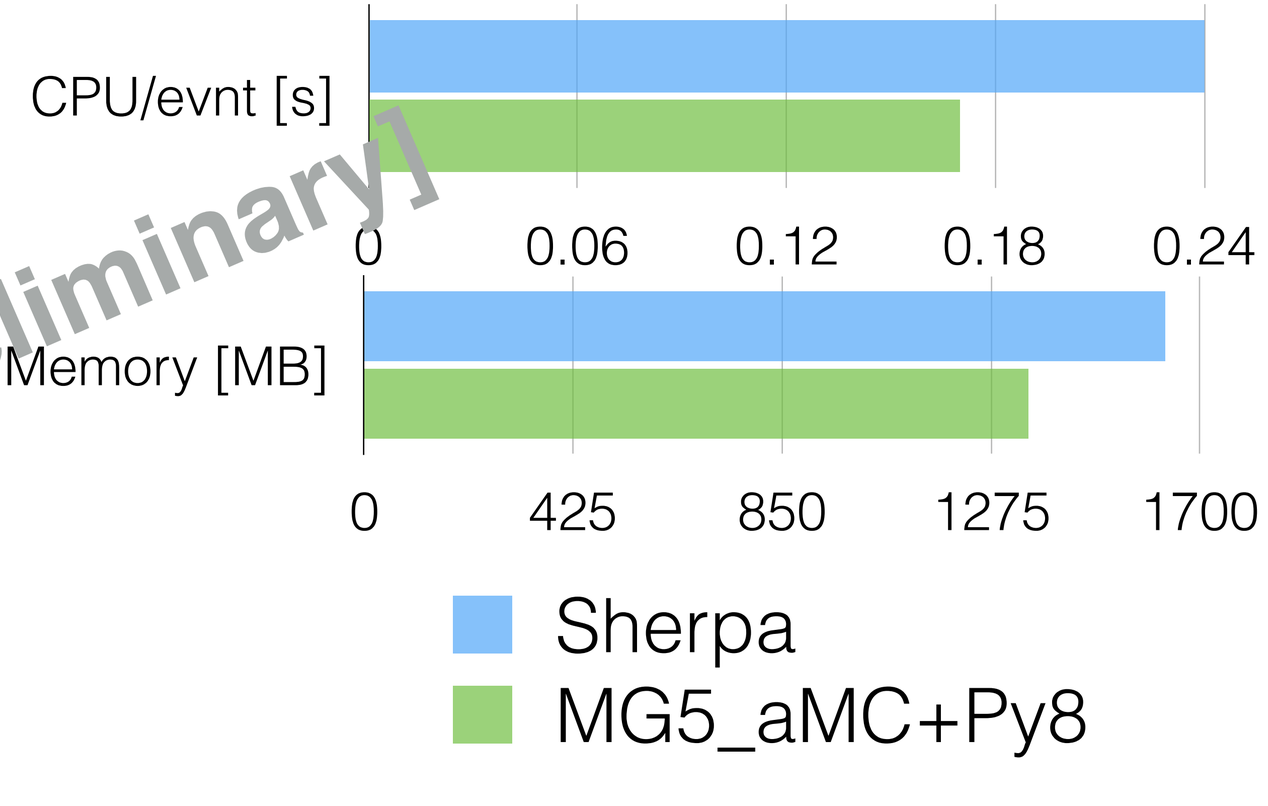
- Sacrifice speed for modelling/precision?
  - Choose the faster generator if there is some big disparity between generators?
    - But what are the speeds?! See next slide for some very preliminary benchmarking.
  - Could consider some form of reweighting lower precision samples to higher precision
    - E.g. LO multileg —> NLO+LO multileg (also works to solve negative weights).

Generator benchmarking | W+jets

W+0-2j@NLO (using pre-integration)



W+0-4j@LO (using pre-integration)

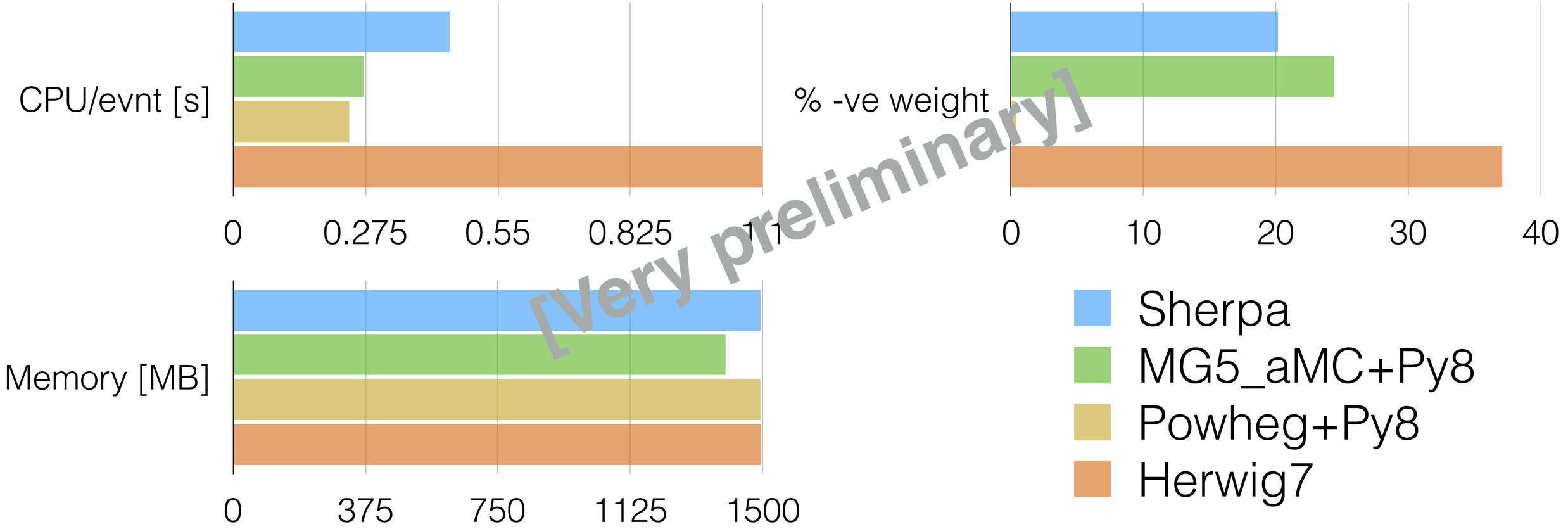


[Very preliminary]

- We are starting to develop apples-to-apples comparisons in ATLAS software framework.
  - All tests run same on “bare metal” machine.
  - 10 x 5-10k event runs are averaged.
- Caveats
  - Not perfectly optimised for matching efficiency.
  - MG5_aMC used internal PDF, not LHAPDF.
  - Memory consumption likely not truly representative.

Generator benchmarking | ttbar

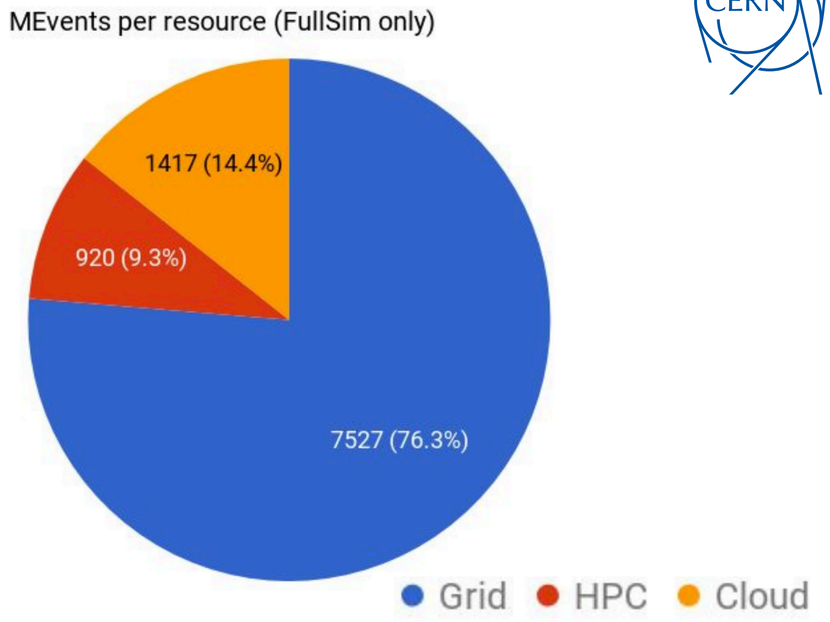
Incl NLO (with on-the-fly integration)

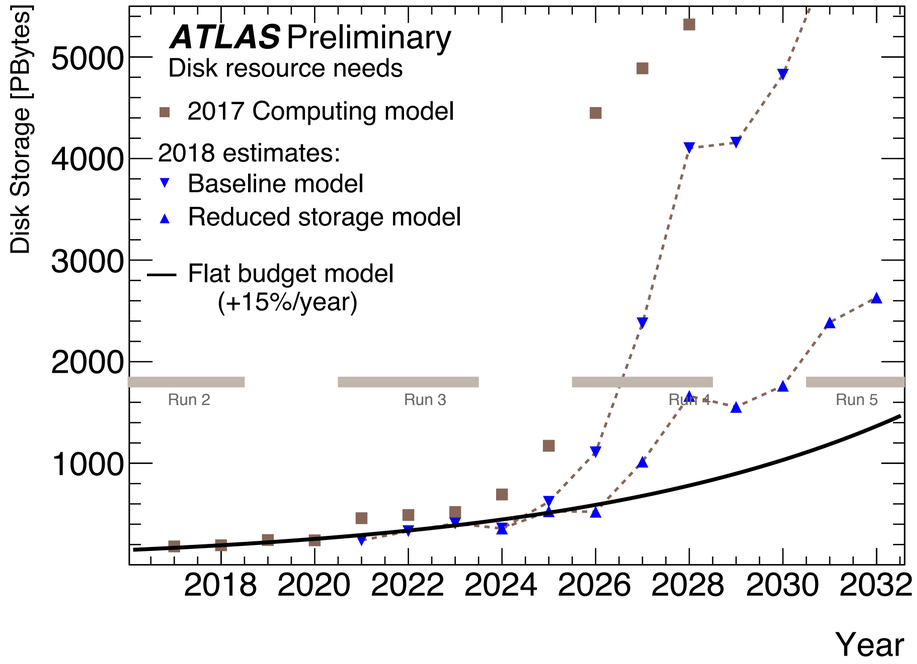


- Further studies:
  - Check w/wo EvtGen/LHAPDF.
  - Check w/wo systematic variation weights.
  - Check with phase-space slicing.
  - More…?
    - We are just starting and very happy to get feedback!

Infrastructure?

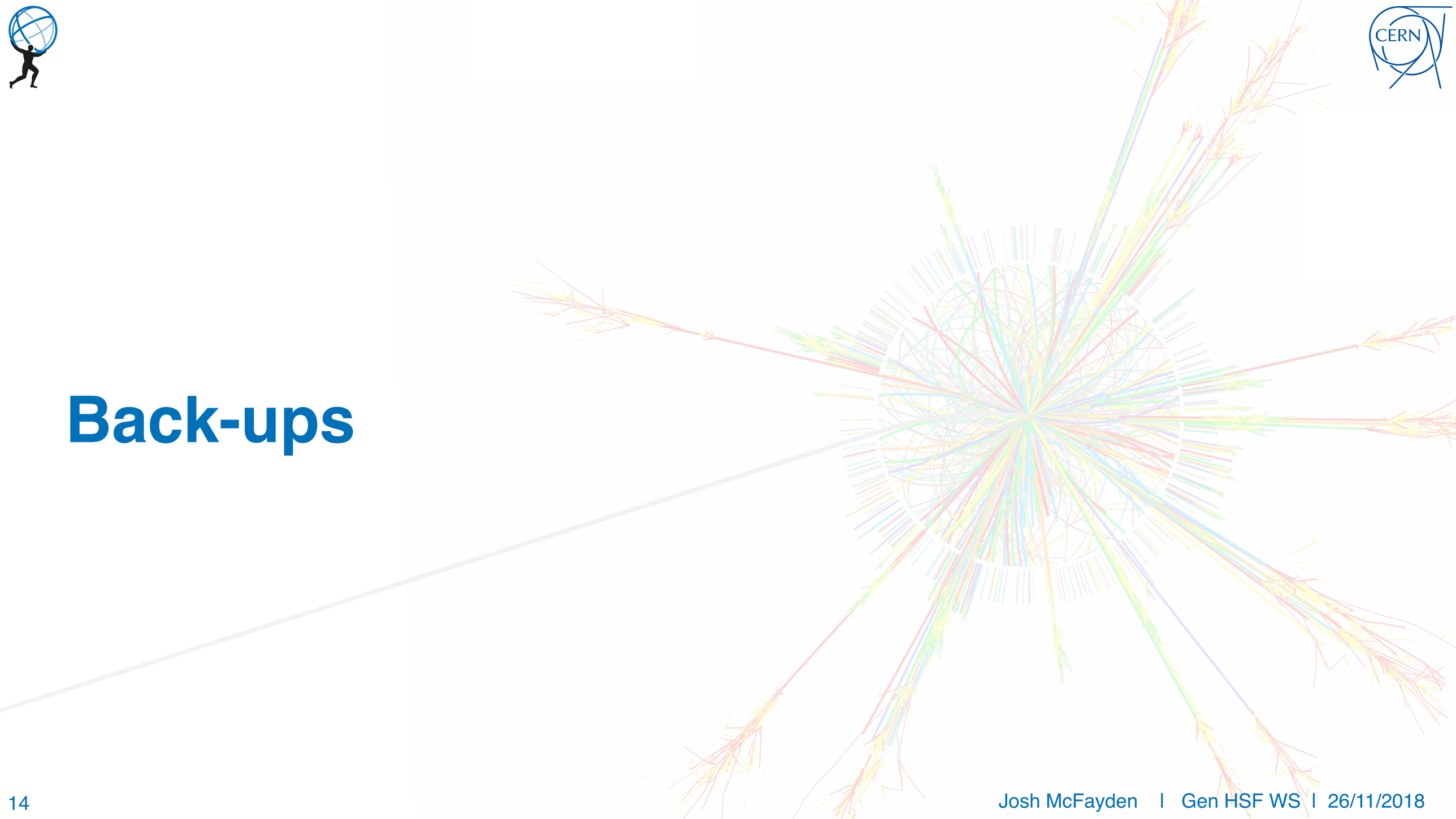
- Possible reversal of strategy ordering:
  - ATLAS/CMS could develop a common computing scheme/framework for MC generation to which the MC collaborations could adapt.
  - More simply, one could ask e.g. “will evgen be predominantly run of CPUs or GPUs in 2028?”
    - Likely driven by next HPCs - need to be flexible?
- ATLAS & CMS sharing samples?
  - Clearly this would involve some complications and likely compromises.
  - But would trivially gain a factor of two in sample size!
- Disk resources are also under severe pressure
  - Likely not a problem unless 100 PBs are written out…





Summary

- Despite many welcome improvements in MC event generator precision and functionality over the last years the current model is not going to scale well…
- There is surely some “low-hanging fruit” for efficiency improvements.
  - Event generation has probably not seen as much scrutiny as other steps until now.
- Optimisation of existing generator codes could surely improve this situation.
  - The best way to achieve this can be discussed during the workshop!
- Efficient optimisation will require advance knowledge of what the computing infrastructures and workflows will look like for HL-LHC.
- Experiments may have to compromise in order to get the MC statistics required.
- For all the above, by starting early (now) hopefully anything is possible!



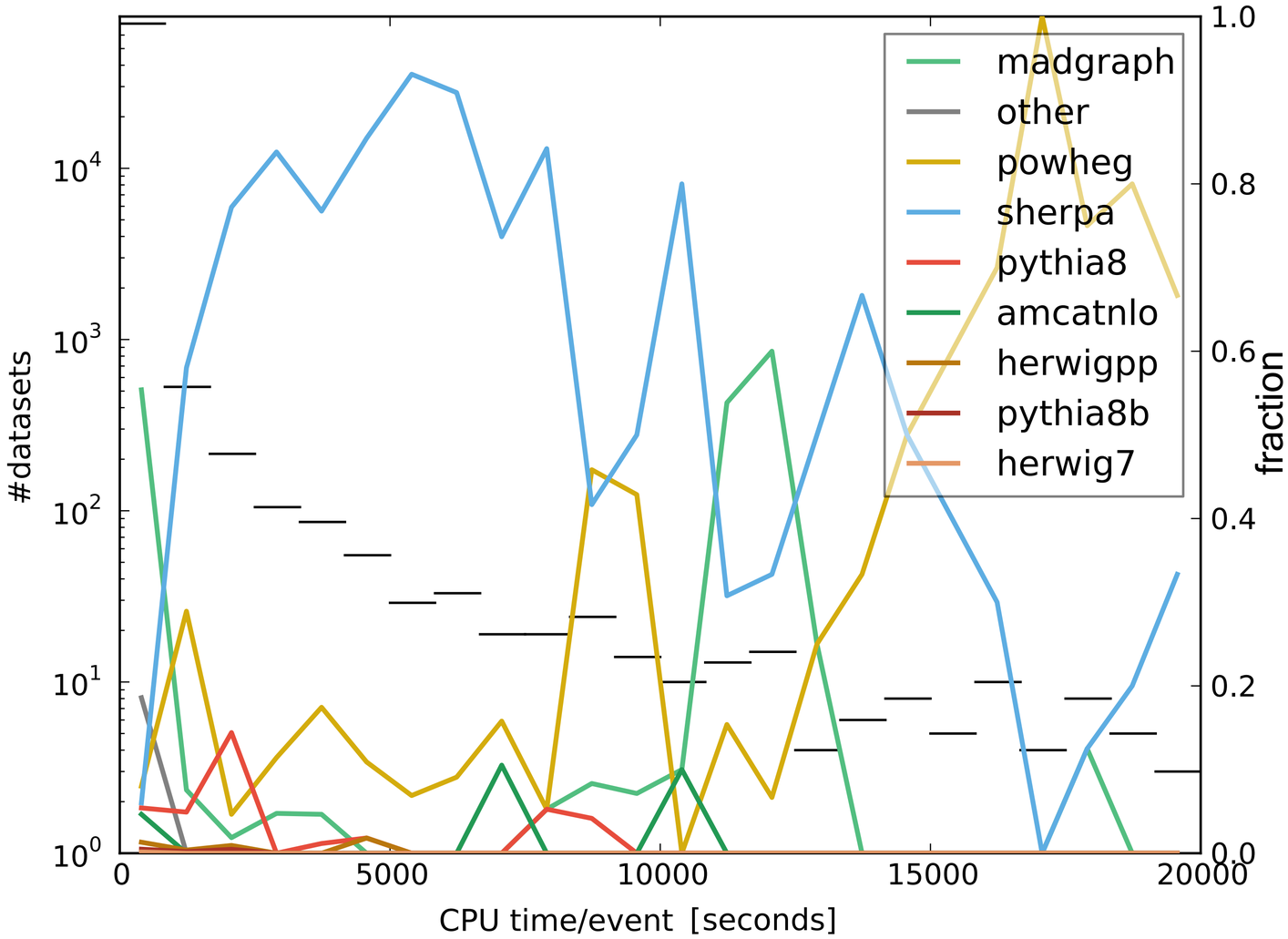
Back-ups

Current generator usage | Avg CPU/event

	Avg CPU/evt
Evgen	~80 s
FullSim	~245 s
FastSim	~45 s
Reco	~60 s

- Average CPU/event by dataset
  - Each entry is a single dataset
  - The numbers have not been corrected for e.g. filter efficiencies
  - ME generators also include showering
    - Sometimes only showering… but it's hard to separate
- The average across all samples is ~80s/event
- But have many examples where event generation is slower than full simulation…
  - A strong indication that this is something to improve!

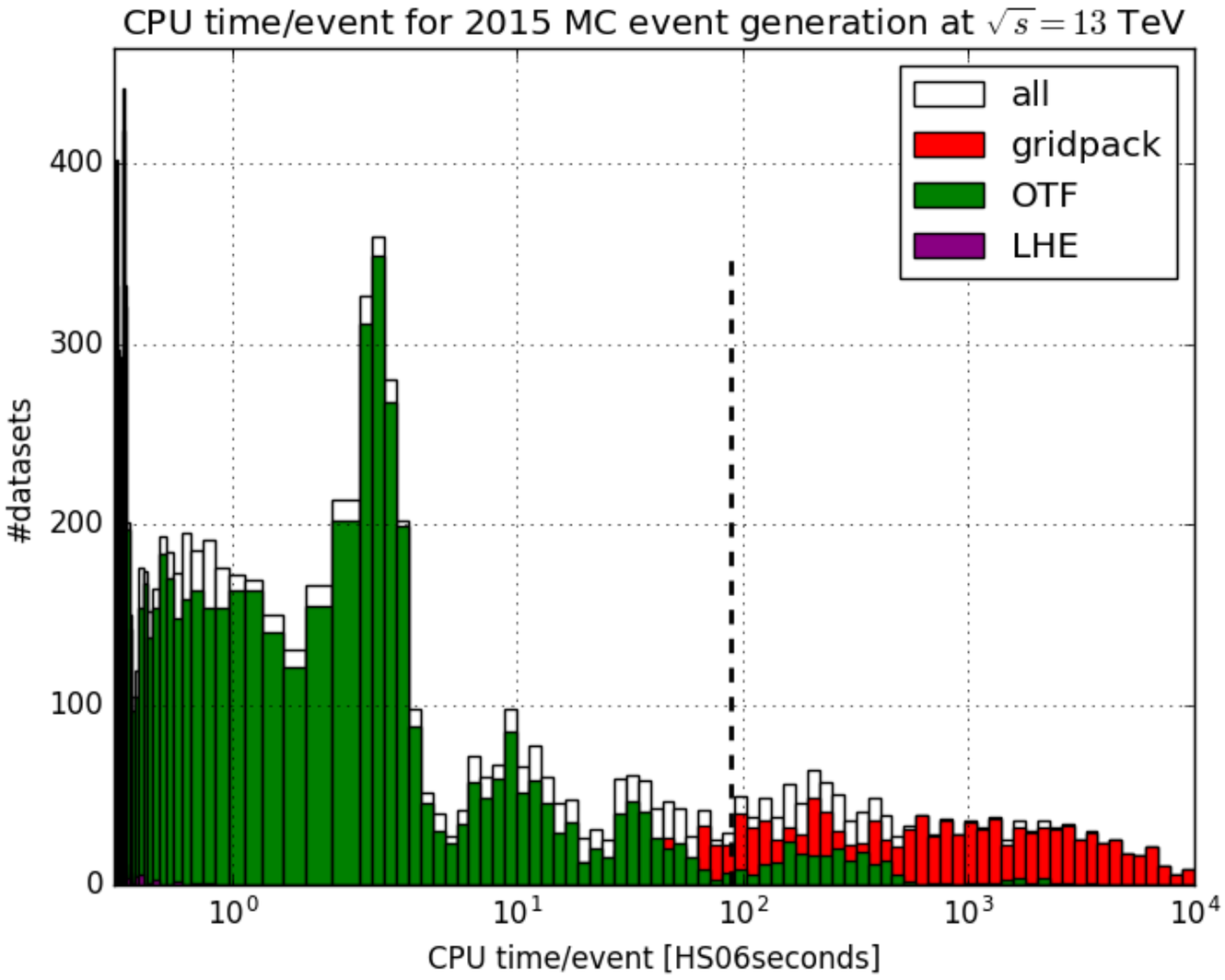
CPU time/event for 2015 MC event generation at $\sqrt{s} = 13$ TeV
(All physics processes included, correlations with process complexity and filter efficiencies not taken into account)



Deedback on possible dedicated post(s)

- In general the proposal was welcomed and agreed that it would be useful.
- Structure
  - Needs international coordination (some independent efforts already in place or starting, e.g. SciDac in US)
  - Being evenly distributed across generator groups is important but may be hard in practise
    - Completely different code structures. Would certainly need supervision from authors.
    - But, no real concern about sharing information...
- Software
  - Authors are aware of code shortcomings but not sufficient time/expertise to improve - would welcome external effort.
  - Possible reversal of strategy ordering:
    - ATLAS/CMS should develop a common computing scheme/framework for MC generation to which the MC collaborations could adapt
  - Possibility for CMS/ATLAS to have a framework which would allow to share common (large) MC sets.
- Physics
  - Negative weight fractions need physics background to solve - process dependant
- People
  - General agreement this will be hard! No consensus on how best to find the right people…
    - To get good candidate these should be long-term posts.
    - Suggestion to make this a "secondment" from experiment with at least 50% FTE spent on these projects (unlikely to come from theory)
      - Externally funded PhD projects under joint supervision between people from Computer Science departments and theory?
      - Graduate student in computer science who wants some experience with real problems.
  - Strong agreement that close interaction/supervision from authors would be necessary
  - General agreement that having two positions, one predominantly on physics-related topics and one on core software would be useful.

Generation type



Integrations

- Ballpark figures for the integration times
  - Run on different machines/clusters etc.

- Completely negligible compared to event generation and showering

- W+0,1,2j@NLO
  - Sherpa = 5h on 96 cores
  - MG5_aMC = 2h on 16 cores

- W+0,1,2,3,4j@LO
  - Sherpa = 3.5h on 96 cores
  - MG5_aMC 8h on 32 cores

Benchmarking

- Intel(R) Xeon(R) CPU E5-2667 v2 @ 3.30GHz