HPC framework for event generation at colliders

Taylor Childers, Stefan Höche, Paul Hovland, Stephen Mrenna,
Stefan Prestel, Holger Schulz

SciDAC PI Meeting 2019
16-18 July, 2019

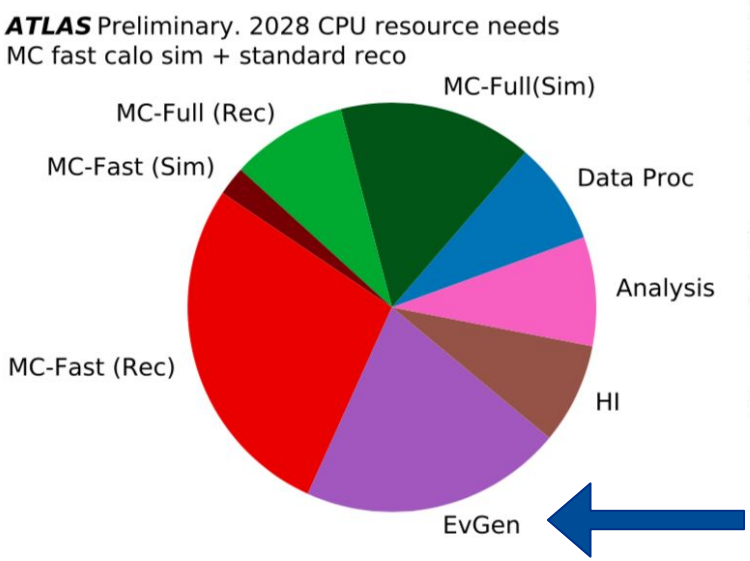
Why improve event generators?

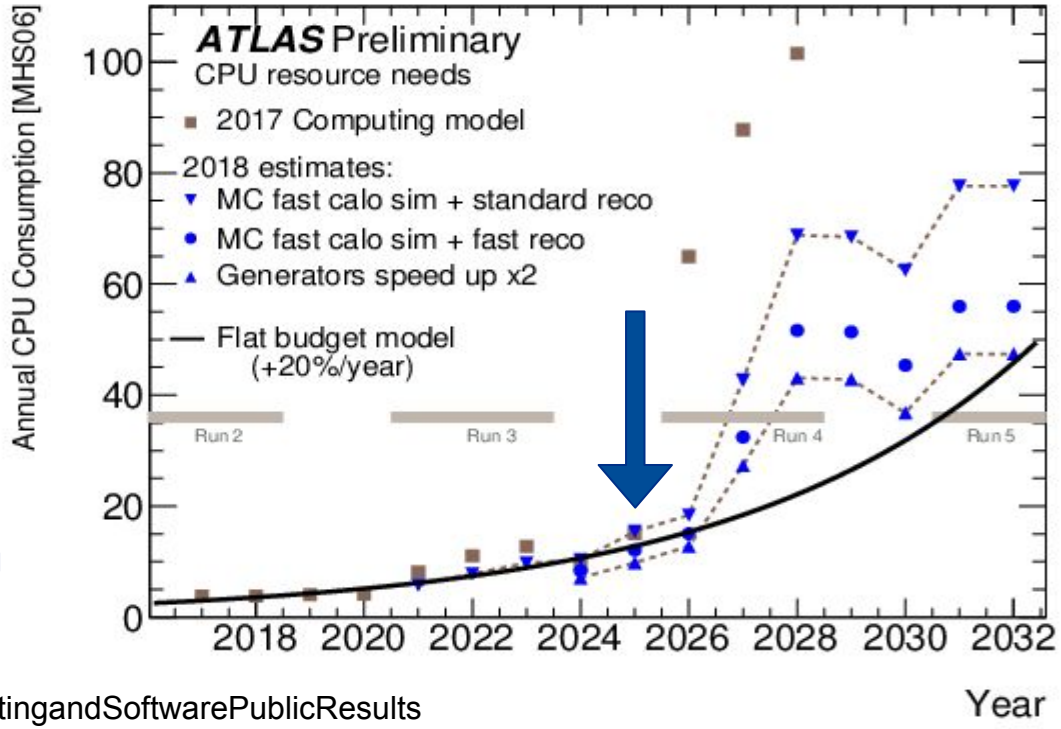
Event generation will consume significant fraction of resources at LHC soon

Need to scrutinize both generator usage and underlying algorithms

Dedicated effort in HEP Software Foundation (HSF)

https://hepsoftwarefoundation.org/workinggroups/generators.html

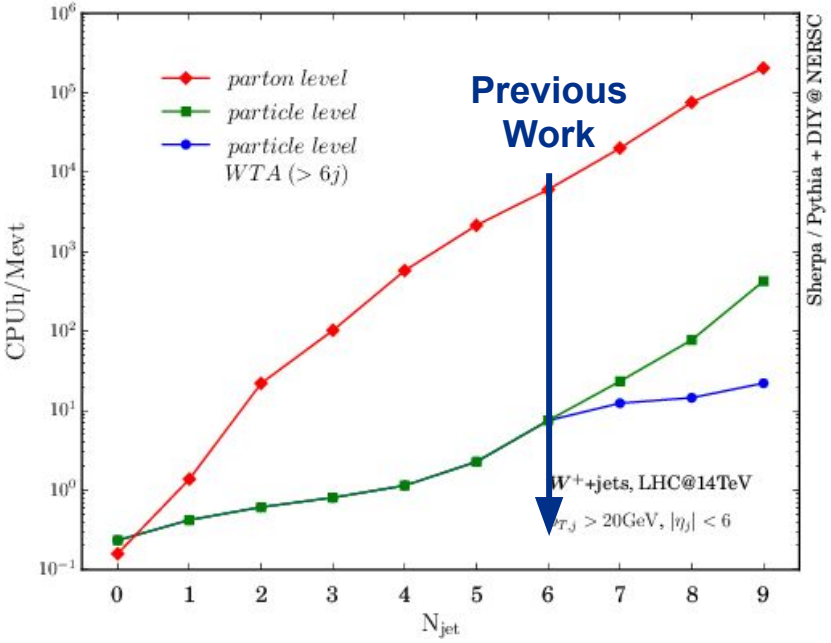


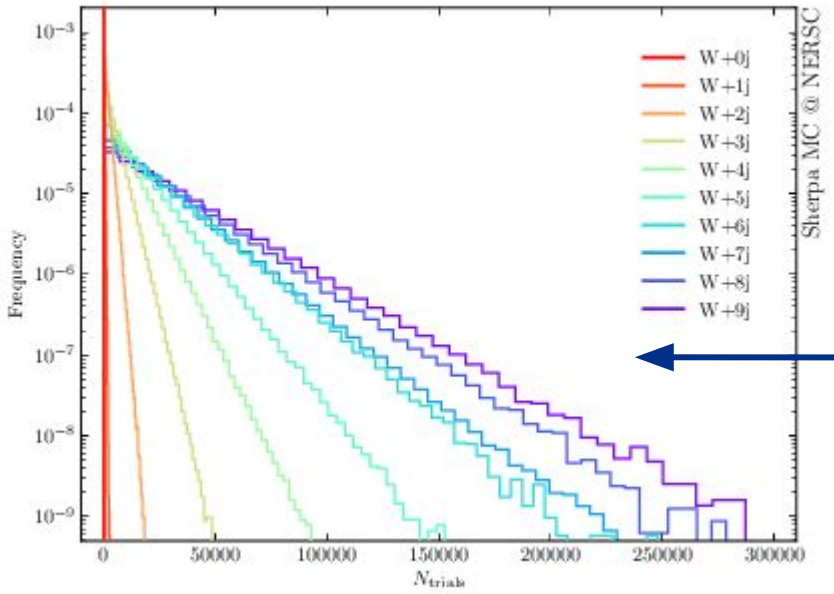


https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ComputingandSoftwarePublicResults

Timing and Scaling

- Hard scattering simulation (ME) much more demanding than parton shower (PS) & hadronization
- Complexity of merging ME&PS grows quickly due to inherent N! scaling of underlying algorithms



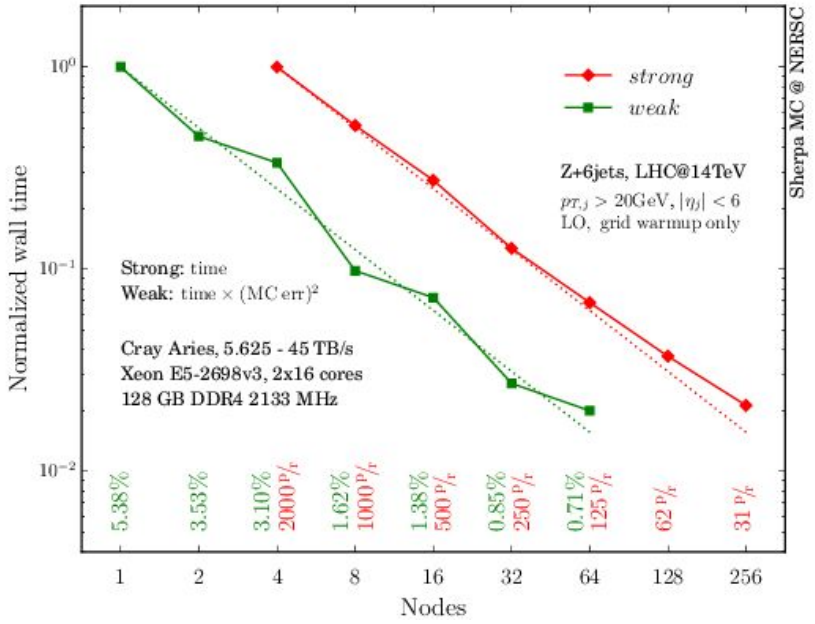


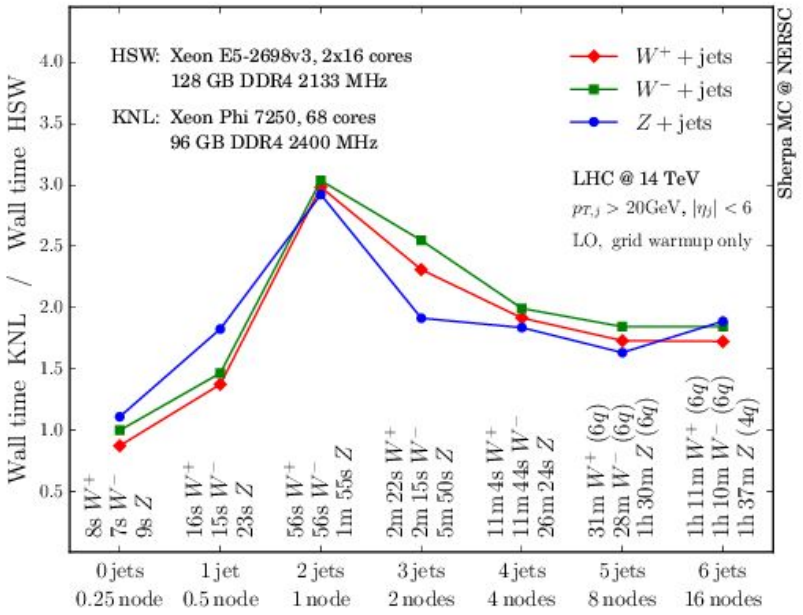
- ME evaluation time naively scales as ~ O(3^N)
- Monte-Carlo unweighting efficiency degrades quickly, as dimensionality of integral is 3N-4
- Overall scaling ~ O(4^N)

Adapting current generators for HPC

- Separate ME generation from PS evolution and ME+PS merging
- Store intermediate status of event in HDF5 files
- Parallelize event processing at particle level using ASCR's DIY

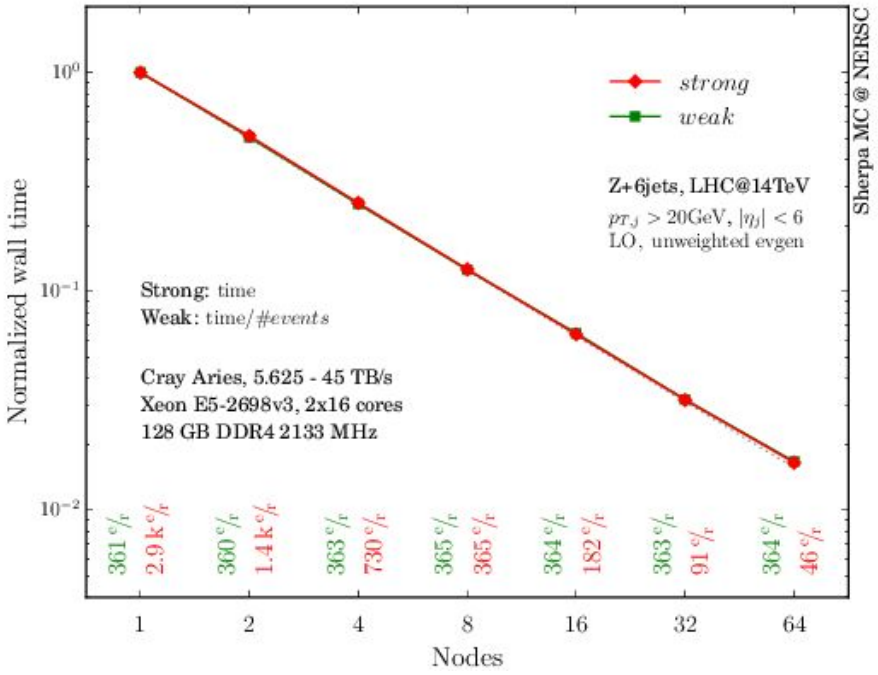
- Performance limited by number of events being processed per rank
- Scaling of optimization step (strong & weak) up to ~ 2048 cores
- Acceptable (though limited) performance on KNL w/o modifications

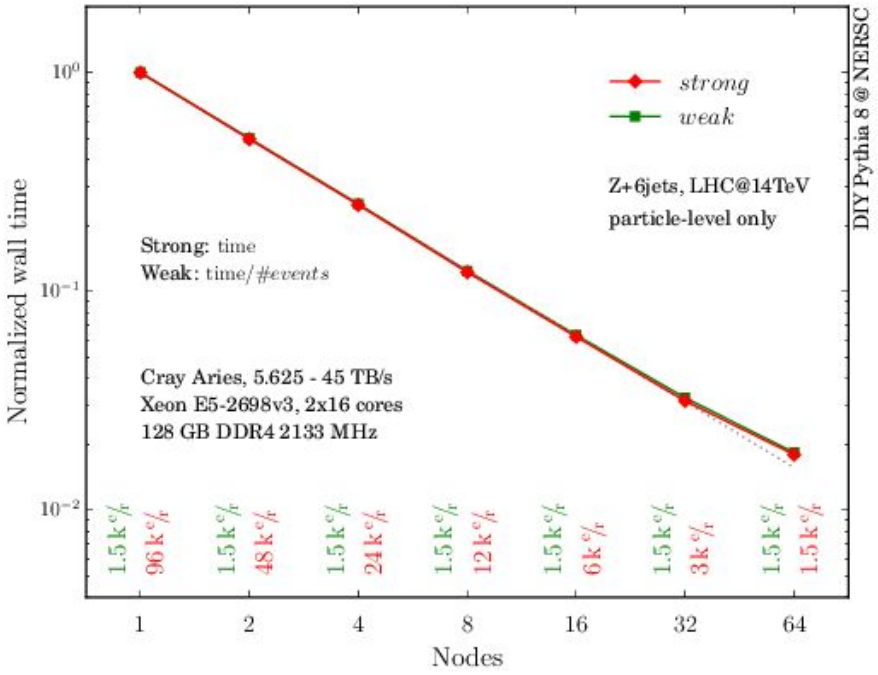




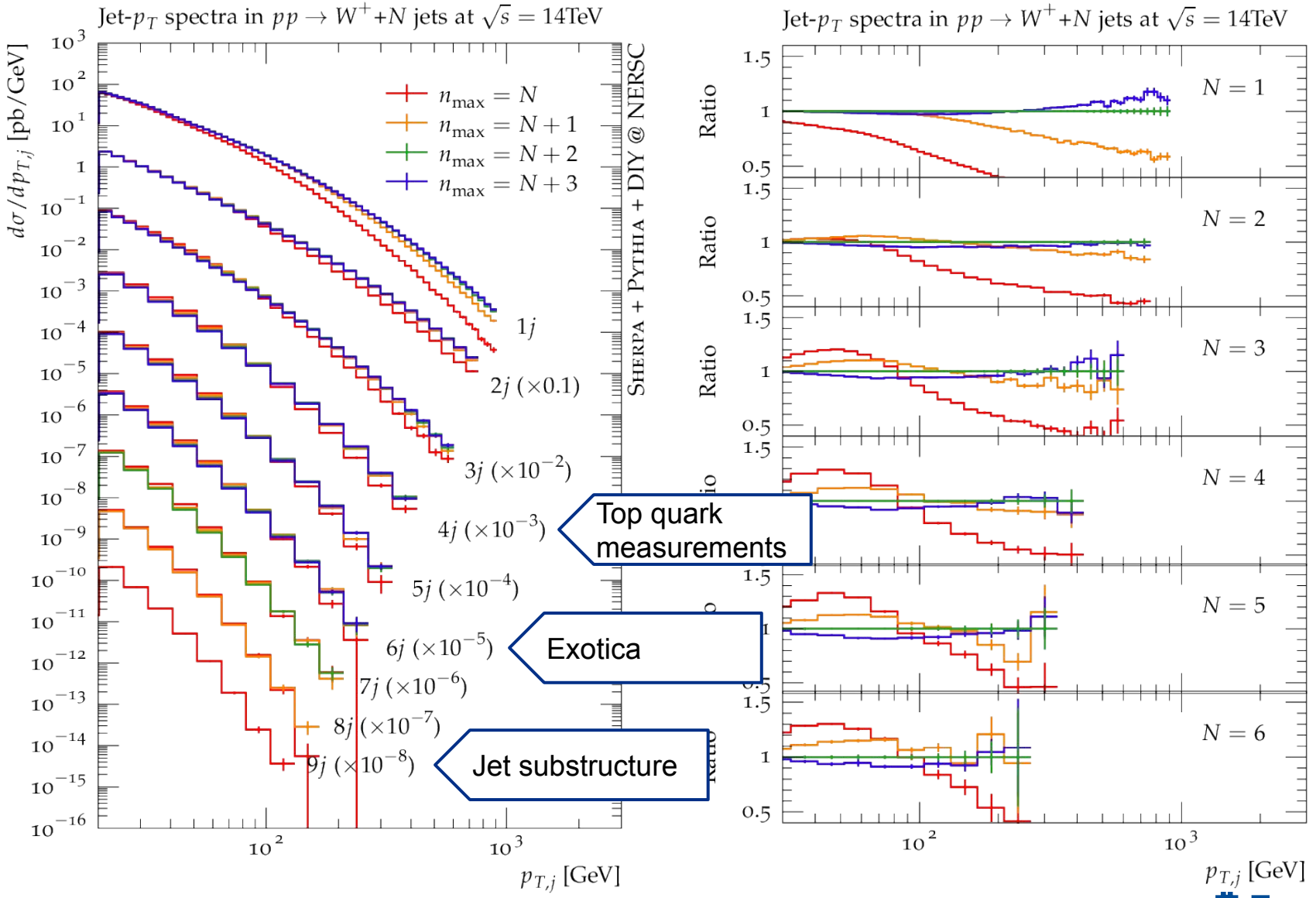
Adapting current generators for HPC

- Scaling of event generation step (strong & weak) up to ~ 2048 cores
- Performance limited by number of events being processed per rank (Average timing can only be expected if statistics is large enough)
- Performance of particle-level simulation limited by I/O speed on Cori Good results with burst buffer, but room for improvement





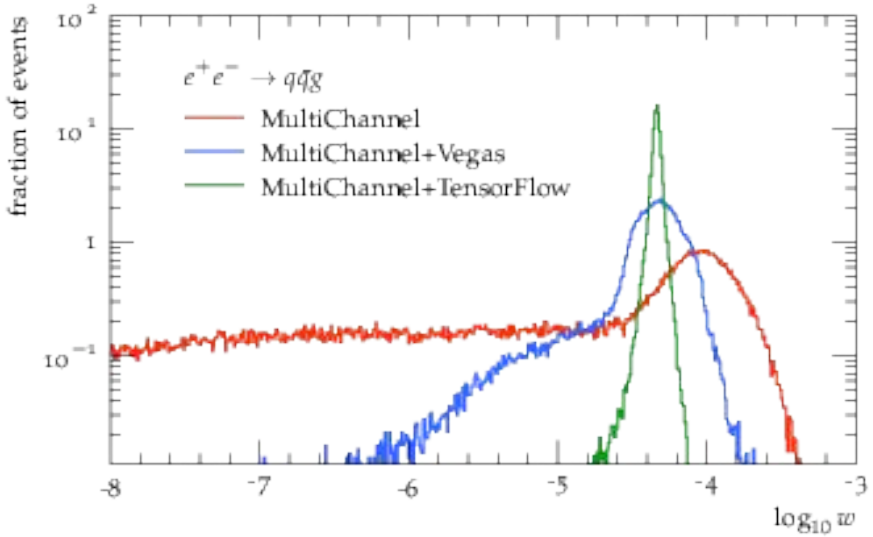
Physics performance



Possible efficiency improvements

Christina Gao, Joshua Isaacson, Claudius Krause, SH

- Testing ideas for novel adaptive integrator, using established MC multi-channeling, combined with neural importance sampling via TensorFlow
- Promising results for simplest non-trivial processes, more complicated cases being tested



- If successful, mandates re-write of matrix-element generator Comix to enable vectorization and support of different architectures
- Possibility to use NumPY / TensorFlow will be explored

Summary and Outlook

Technical improvements

- Parallelized Pythia particle-level event generation using DIY
- Improved performance of Sherpa, particularly I/O (HDF5)
- Optimization & event generation now scale up to ~2k cores
- First novel results not previously attainable on WLCG

Physics impact

- New workflow allows to reduce EvGen time in experiments
- Novel opportunity of generator tuning based on theory

  Cross-cut with SciDAC “HEP Data Analytics on HPC”
- Novel opportunities for jet substructure analyses