Physics Event Generator Computing Workshop

Summary from GEN

Jan 24th, 2019

Efe Yazgan and Qiang Li

Selected and Biased review. Not meant to be exhaustive.

中国科学院高能物理研究所
Institute of High Energy Physics
Chinese Academy of Sciences

PEKING UNIVERSITY 1898

Workshop Introduction

The HEP Software Foundation facilitates cooperation and common efforts in High Energy Physics software and computing internationally.
Community White Paper： summarising R&D in a variety of technical areas for HEP Software and Computing

HEP Software Foundation

Physics Event Generator Computing Workshop

26 Nov 2018, 09:00 → 28 Nov 2018, 18:00 Europe/Zurich

4-3-006 - TH Conference Room (CERN)

Goals of this workshop:

○ Identify the most crucial areas for technical improvements to the generators used by the experiments.

○ Define a programme of work that can be used to attract investment in these technical areas, aiming to have software engineers who can work together with the generator authors.

○ Identify ways of making new theoretical advances easier to implement in a computationally efficient way.

Workshop Agenda

Addressing the Challenges - Day 1 Morning

Given the mixed audience we thought we'd have an experimental introduction and a generator intro. describing the different moving parts of generator codes.

These talks will focus on experiment requirements. For example, what is needed in terms of accuracy & statistics so as not to be limited by MC?

Monday, 26 November

- **09:00 → 09:05** Welcome and Practicalities
  Speakers: Andrea Valassi (CERN), Graeme Stewart (CER
- **09:05 → 09:30** CWP Challenges and Workshop Aims
  We review the challenges for the HEP field with an empha
  Speakers: Elizabeth Sexton-Kennedy (Fermi National Acc
- **09:30 → 10:00** Introduction to Generator Codes
  Speaker: Simon Platzer (University of Vienna (AT))
- **10:00 → 10:20** ATLAS Needs and Concerns
  Speaker: Josh McFayden (CERN)
- **10:20 → 10:40** CMS Needs and Concerns
  Speaker: Efe Yazgan (Chinese Academy of Sciences (CN
- **10:40 → 11:10** Coffee
- **11:10 → 11:30** LHCb Needs and Concerns
  Speaker: Philip Ilten (University of Birmingham (GB))
- **11:30 → 11:50** Practical Computing Considerations
  Speaker: David Lange (Princeton University (US))
- **11:50 → 12:15** Discussion
  25m 4-3-006 - TH Conference Ro...

Workshop Agenda

Addressing the Challenges - Day 1 Afternoon

Here we hear in more detail about the technical requirements on modern codes, suitable for use on HPCs.

Status and plans of the widely used generators. We hope to have a focus on algorithms and performance. Is it possible to define an agreed upon benchmark physics process?

12:15 → 13:15	Lunch	1h	4-3-006 - TH Conference Ro...
13:15 → 13:45	Performance Optimisation Introduction Speaker: Servesh Muralidharan (CERN)	30m	4-3-006 - TH Conference Ro...
13:45 → 14:15	Optimising Memory Use Speaker: Sebastien Ponce (CERN) MemoryUsageO...	30m	4-3-006 - TH Conference Ro...
14:15 → 15:00	Performance Discussion	45m	4-3-006 - TH Conference Ro...
15:00 → 15:30	Coffee	30m	4-3-006 - TH Conference Ro...
15:30 → 15:53	Sherpa Status and Plans Speaker: Marek Schoenherr (CERN)	23m	4-3-006 - TH Conference Ro...
15:53 → 16:16	Madgraph Status and Plans Speaker: Olivier Mattelaer (UCLouvain)	23m	4-3-006 - TH Conference Ro...
16:16 → 16:38	PYTHIA 8 Status and Plans Speaker: Philip Ilten (University of Birmingham (GB))	22m	4-3-006 - TH Conference Ro...
16:38 → 17:00	Herwig Status and Plans Speaker: Peter Richardson (CERN & IPPP, Durham University)	22m	4-3-006 - TH Conference Ro...
17:00 → 17:20	POWHEG Status and Plans Speaker: Emanuele Re (CERN)	20m	4-3-006 - TH Conference Ro...
17:20 → 18:00	Discussion	40m	4-3-006 - TH Conference Ro...

Workshop Agenda

Addressing the Challenges - Day 2 Morning

Reports of ongoing work to make generator codes more suitable for use on HPC systems.
Can these ideas be made suitable for many generators?

discussion Event reweighting? Sharing between ATLAS & CMS ?

Status/plans for generators used as "decayers"

Tuesday, 27 November

09:00	→ 09:25	**HDF5 usage for HEP event generation** Speaker: Holger Schulz (Fermilab)	25m	4-3-006 - TH Conference Ro...
09:25	→ 09:50	**A novel workflow of generator tunings in HPC for LHC new physics searches** Speaker: Xiangyang Ju (Lawrence Berkeley National Lab. (US))	25m	4-3-006 - TH Conference Ro...
09:50	→ 10:15	**Adaptive multi-channel integration with MPI** Speakers: Simon Brass (University of Siegen), Simon Braß (Universität Siegen)	25m	4-3-006 - TH Conference Ro...
10:15	→ 10:30	**Discussion**	15m	4-3-006 - TH Conference Ro...
10:30	→ 11:00	**Coffee**	30m	
11:00	→ 12:00	**Optimising Generator Useage** Speakers: Andy Buckley (University of Glasgow (GB)), Marek Schoenherr (CERN), Stefan Hoeche (SLAC)	1h	4-3-006 - TH Conference Ro...
12:00	→ 12:15	**Photos and Tauola Status and Plans** Speaker: Zbigniew Andrzej Was (Polish Academy of Sciences (PL))	15m	4-3-006 - TH Conference Ro...
12:15	→ 12:30	**EvtGen Status and Plans** Speaker: Michal Kreps (University of Warwick (GB))	15m	4-3-006 - TH Conference Ro...

Workshop Agenda

Day 2 Afternoon and Day 3

NNLO...

Beyond current practice

Follow up planning

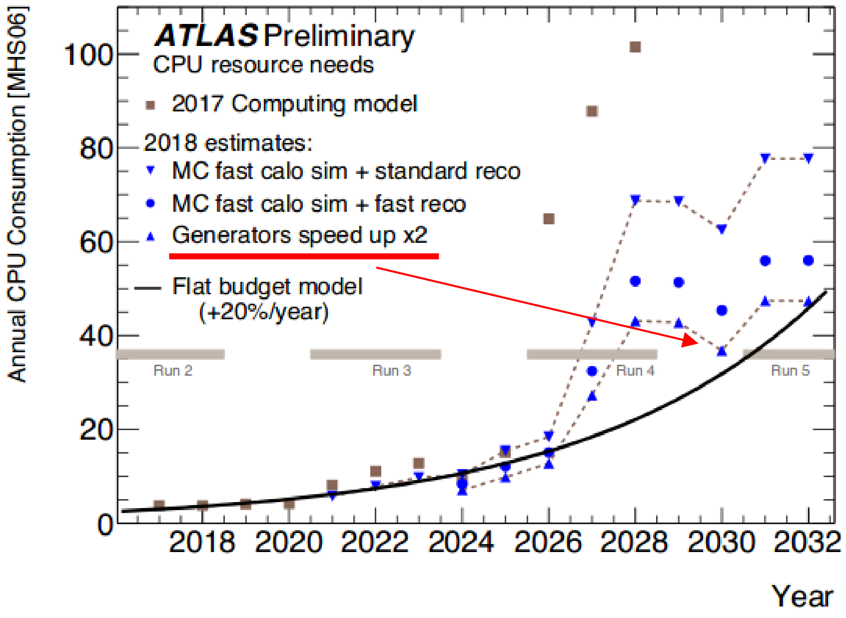
Hackathon, sign up here:
https://docs.google.com/spreadsheets/d/1uReZEVU-wS0liNiqaRjci7fPrJpBPtzrfaa9CKfTbTU/edit#gid=0

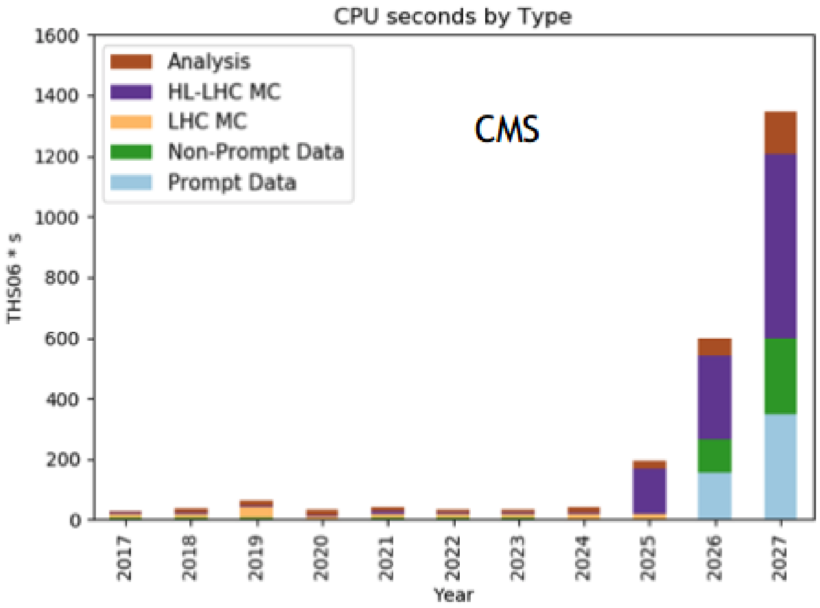
12:30 → 13:30	Lunch	1h	
13:30 → 13:55	NNLO / N^3LO calculations with NNLOJet Speaker: Alexander Yohei Huss (CERN)	25m	31-3-004 - IT Amphitheatre
13:55 → 14:20	NNLO calculations with Matrix Speaker: Marius Wiesemann (CERN)	25m	31-3-004 - IT Amphitheatre
14:20 → 14:45	NNLO calculations with MCFM Speaker: Tobias Neumann (Illinois Tech / Fermi National Accelerator Laboratory)	25m	31-3-004 - IT Amphitheatre
14:45 → 15:00	Discussion	15m	31-3-004 - IT Amphitheatre
15:00 → 15:30	Coffee	30m	
15:30 → 15:50	Beyond Current Paradigms Speaker: Simon Platzer (University of Vienna (AT))	20m	31-3-004 - IT Amphitheatre
15:50 → 17:00	Overflow Discussion, Closeout and Planning	1h 10m	31-3-004 - IT Amphitheatre

Wednesday, 28 November

09:00 → 12:30	Hackathon - Morning Session Speakers: Graeme Stewart (CERN), Servesh Muralidharan (CERN)	3h 30m	5-1-015
12:30 → 14:00	Lunch	1h 30m	
14:00 → 17:30	Hackathon - Afternoon Session Speakers: Graeme Stewart (CERN), Servesh Muralidharan (CERN)	3h 30m	5-1-015

Motivation

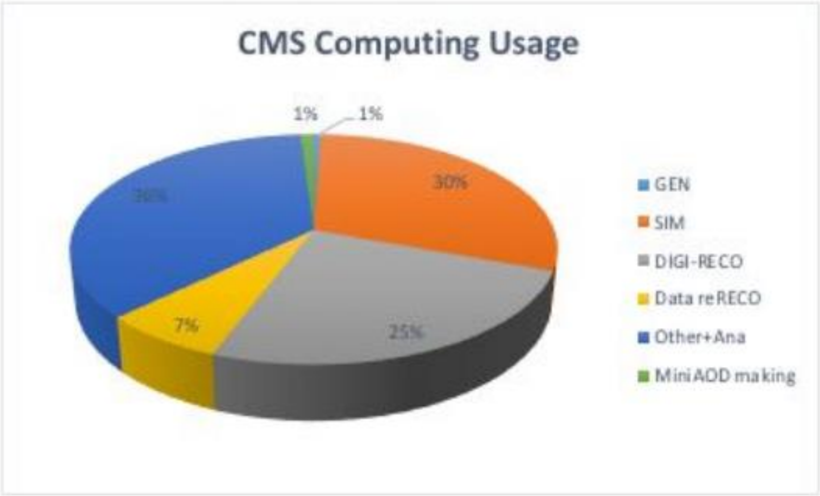


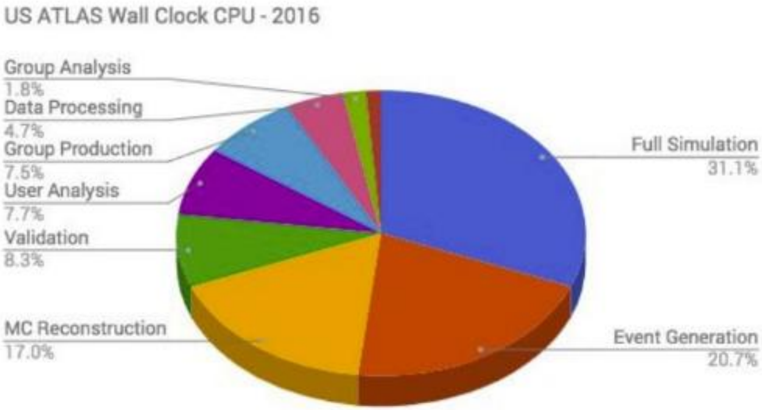


- We are OFF by ~5x on CPU power when considering Moore's law
- HL-LHC salvation will come from software improvements, not from hardware

Elizabeth Sexton-Kennedy

Generators CAN be Computationally Intensive





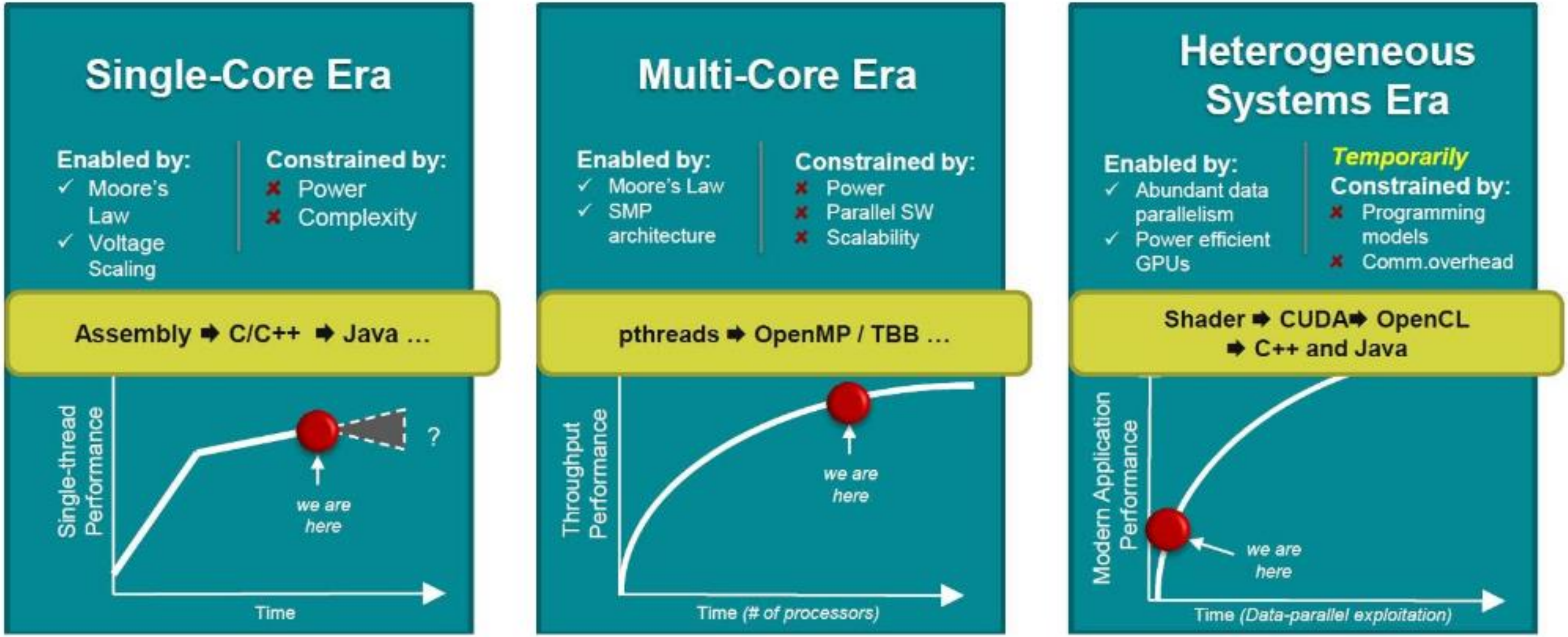
CMS usage from 2017, ATLAS went down to 14.3% in 2017

Elizabeth Sexton-Kennedy

- These values vary from year to year as analysis needs vary
- CMS uses more LO samples in this year and grid-pack configurations

Most HEP tools Typically executes 1 instruction at a time (per thread)

The Future : Heterogeneous Architecture



https://opensourceforu.com/2016/12/how-heterogeneous-systems-evolved-and-the-challenges-going-forward/

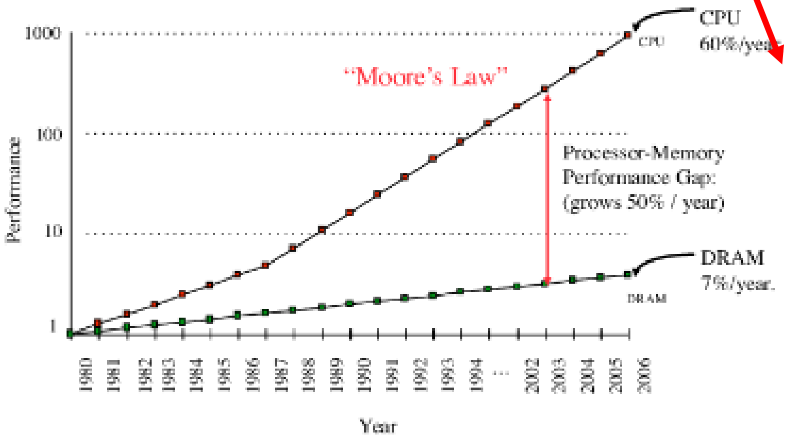
The computing available in 2026 will be heterogeneous and highly concurrent.

different types of compute units and interconnects

Computing Requirements

1. Many Core threading models
2. Single Instruction Multiple Data (SIMD) vectorization
3. Non-uniform Memory Access (NUMA) hierarchies
4. Offloading to accelerators like
   a. Graphic Processing Unit (GPU),
   b. Field Programmable Gate Array (FPGA),
   c. and Tensor Processing Unit (TPU)

[Servesh Muralidharan](#)
[Sebastien Ponce](#)



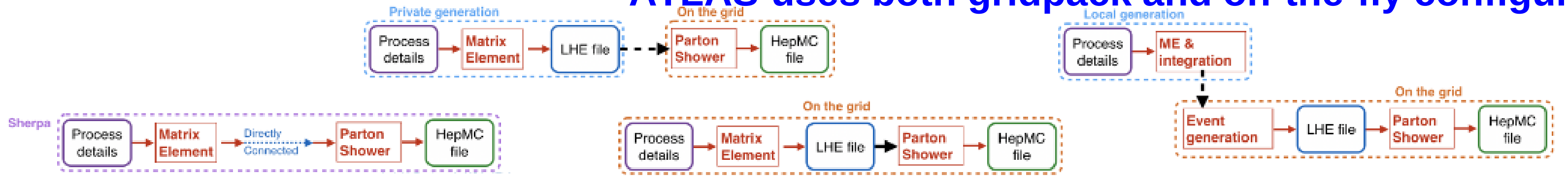
Due to Moore's law in the 80s and 90s, there is a gap between CPU and memory performances

ATLAS Generator Usage

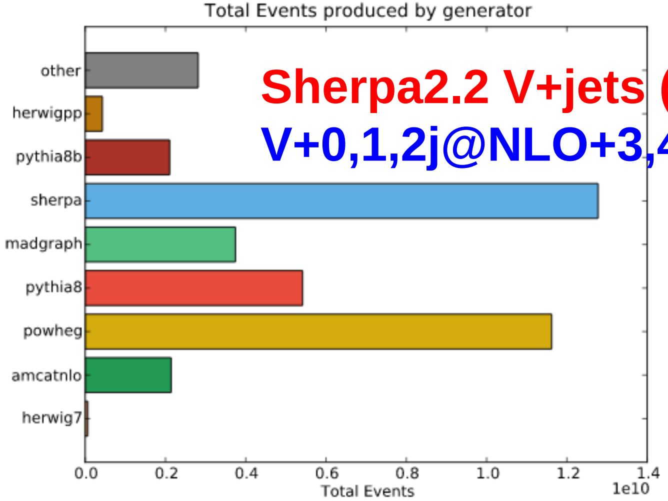
Josh McFayden

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

ATLAS uses both gridpack and on-the-fly configurations

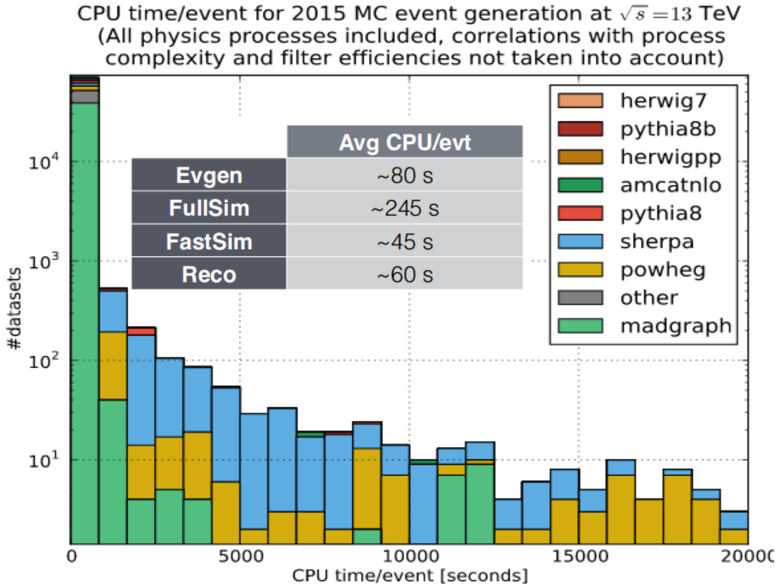


4 Josh McFayden | Gen HSF WS | 26/11/2018



Sherpa2.2 V+jets (3.2B)

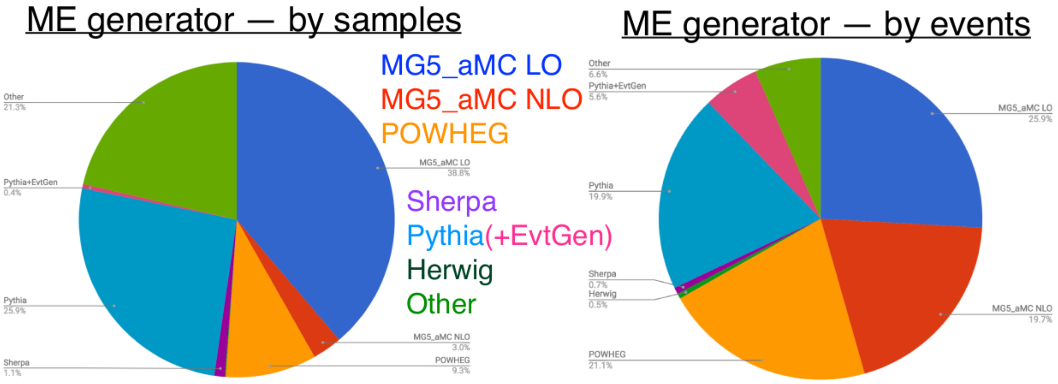
V+0,1,2j@NLO+3,4j@LO

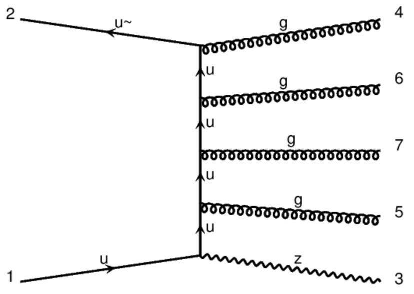


	Avg CPU/evt
Evgen	~80 s
FullSim	~245 s
FastSim	~45 s
Reco	~60 s

Generator usage in CMS [Efe Yazgan]

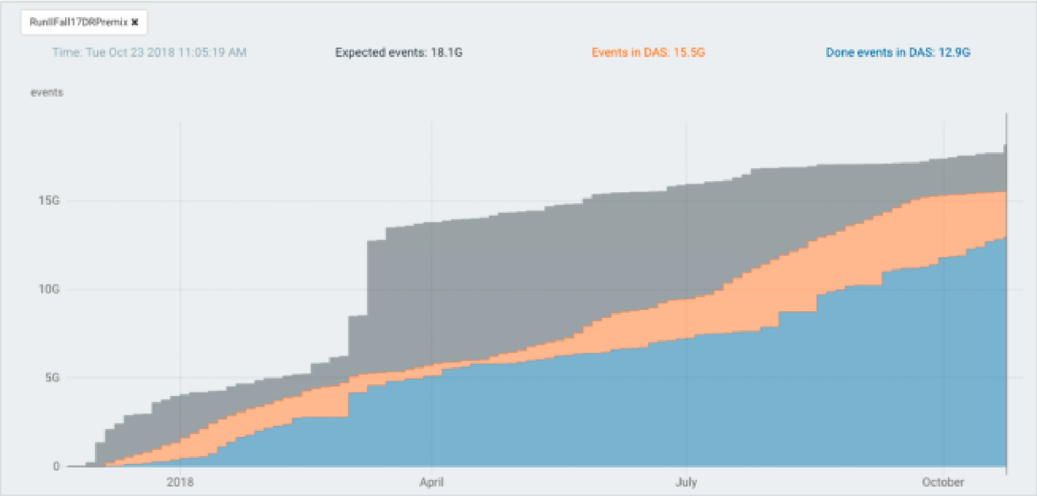
based on 2016 MC campaign





- **Multi-leg LO and NLO consistently matched to the parton shower**
  - **LO: Z+0/1/2/3/4 Jets**
    Most commonly used in CMS: MG5_aMC@NLO+Pythia8 with MLM matching
    Most complex process up to 4 additional jets
  - **NLO TTbar+0/1/2 Jets**
    Most commonly used in CMS: MG5_aMC+Pythia8 with FxFx merging
    Most complex process up to 2 additional jets at NLO.
- **For signal, NNLO+PS**
  POWHEG: MINLO_NNLOPS
  CMS HWW reweight the nominal signal to this one

CMS RunII GEN Production Efe Yazgan



Run II:
→ GEN not stored for physics samples in disk. GEN-SIM re-produced whenever needed.
→ Generators ~1-10% of the total CPU
  → Variation due to LO, NLO, NNLO, complexity of the process, or different methods of calculation.
  → Most BSM samples at this point are simulated at LO.

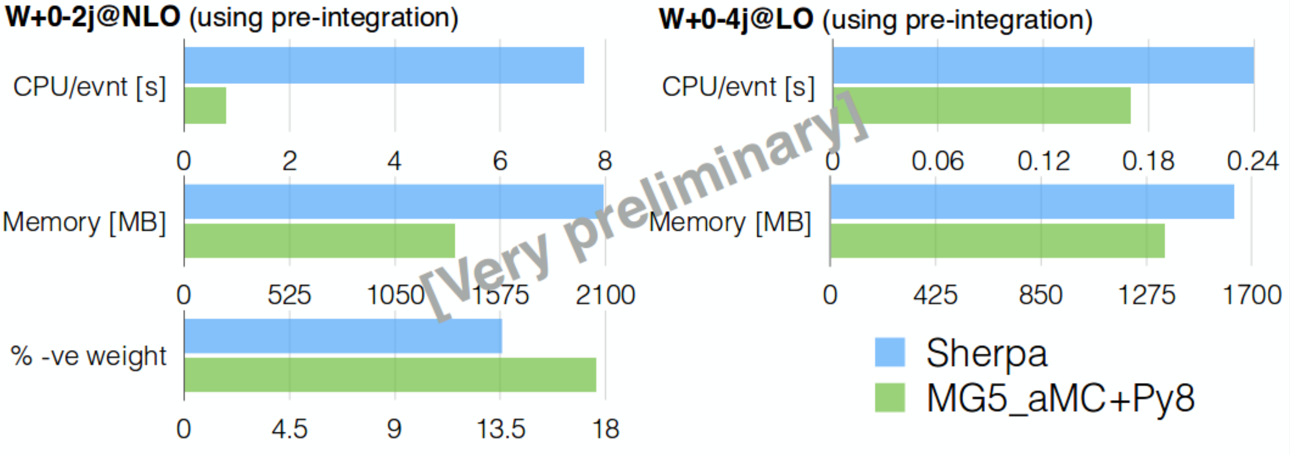
- 15 B (+ some other production campaigns ~ 20 B) in 8 months
  - GEN-SIM-DIGI-RECO ~85 sec/evt
  - 60k cores (~1/3 of the CMS production power)
- Multi-leg LO
  - up to ~10s/gen-evt
  - ~10% matching efficiency → 100s/full-sim-evt
- Multi-leg NLO
  - up to ~30s/gen-evt
  - ~30% matching efficiency → 100s/full-sim-evt
  - Large fraction of negative weights of up to ~40% → larger samples!

8

Performance Comparison

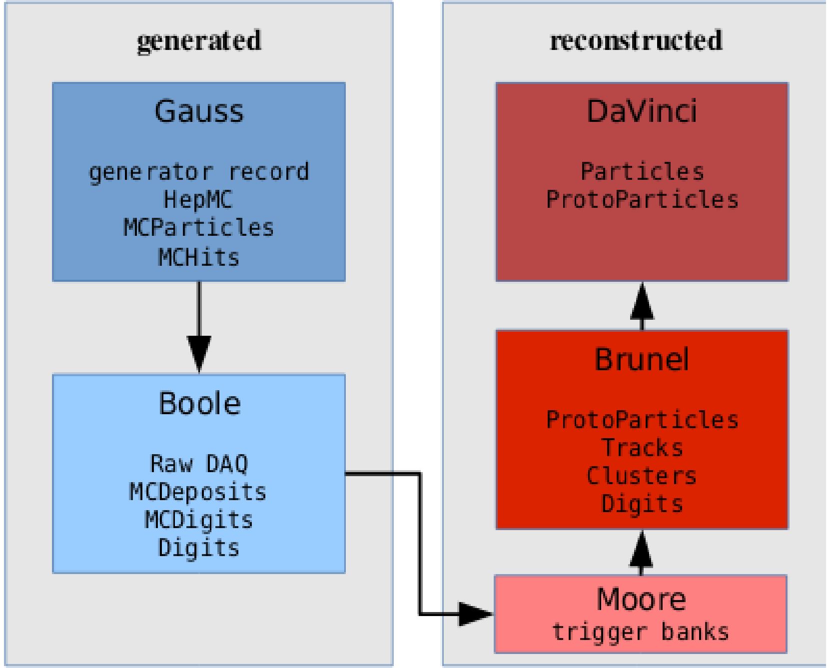
- Physics Event Generator Computing Workshop on 26-29 November.
  - To get a fair comparison of Sherpa vs madgraph vs other leading generators to understand the computing performance.
- The CMS Sherpa contacts have already started it. **Anyone is welcome to join the effort!**
  - Frank Siegert - ATLAS contact
  - Gurpreet Singh Chahal - CMS contact
  - Olivier Mattelaer - MG5 contact
  - Stefan Hoeche - Sherpa contact
    - We will cross check the cards, etc. to help to get these studies done.

Very Prelimenary numbers from CMS in this link: CMS



Josh McFayden

LHCb Generator Usage



- current simulation cannot cope with LHCb Run 3 needs within available CPU resources
- implemented:
  - *filtered events* with fully reconstructed signals introduced, saves space but not time
  - *multiple trigger conditions* per event now stored, saves both time and space
  - *redecay* signal multiple times and reuse fully simulated remaining event [arXiv:1810.10362]
  - *particle gun events* used to produce specific backgrounds and signals, but has limited use
- in progress:
  - fully multi-threaded environment
  - detector geometry simplification and parametrization
  - faster MPI generation?

Philip Ilten

Grid Computing Comparison

One approach use to create Monte Carlo simulation on the grid

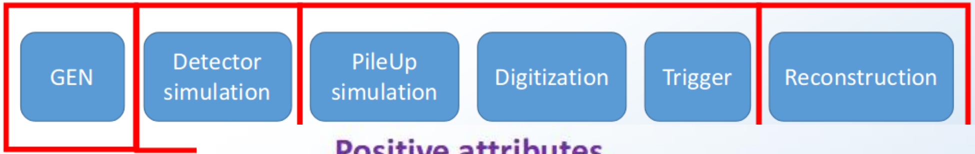
David Lange

Positive attributes

- Generator small piece of total GEN+SIM processing time (historically)
- Less data handling and fewer processing steps

Negative attributes

- Difficult to accommodate generator developments need more agility than simulation (which prefers stability)
- Long startup times in generators (eg, gridpack handling, once-per-job calculations) have bigger impact on throughput



Positive attributes

- Easy to include newest generator developments (assuming stable file format for output)
- Job requirements (cores, resources, etc) can be optimized for generator being used

Negative attributes

- Generators must run many events per job to match grid requirements
- More data handling
- Potentially more latency in processing MC events

CMS Example 1: MadGraph LO Thread Problem

- Christmas computing issue (Madgraph workflow causing issues at HPC sites including NERSC, IRFU, Colorado) on MG LO MLM (consumes as many threads as the machine had processors).
- More details can be found from the GEN Operator Report
- MG Authors only recommend to go for newer version of MG.
- Several Options tried
  - **(1)** **1st** new request (set with run_mode =2 to avoid over consuming threads). It seems ok with ~617 errors out of 30000 jobs. A gridpack patch is ready
  - **(2)** **2nd** new request with mg261 gridpack sees lots of errors at HPC.
  - (3) creating ad-hoc campaigns (duplicating existing wmLHE ones) to process only Madgraph LO requests; such campaigns would black-list the affected sites
  - (4) Block MG LO in the existing wmLHE MC campaigns to non-HPC sites: JIRA

Todo: Update MG260 to 263+ will take some time (validation, check patches); Will keep in contact with the authors; **It would be important to have dedicated manpower to work on this kind of task!**

CMS Example 2: Herwig7 Reval Timeout

- **Workflow 511.1:** pp->e+e- NLO from H7+Openloops, details from Sanghyun Ko
  However, time out due to no integration initiation (i.e. need Herpack to be available soon)

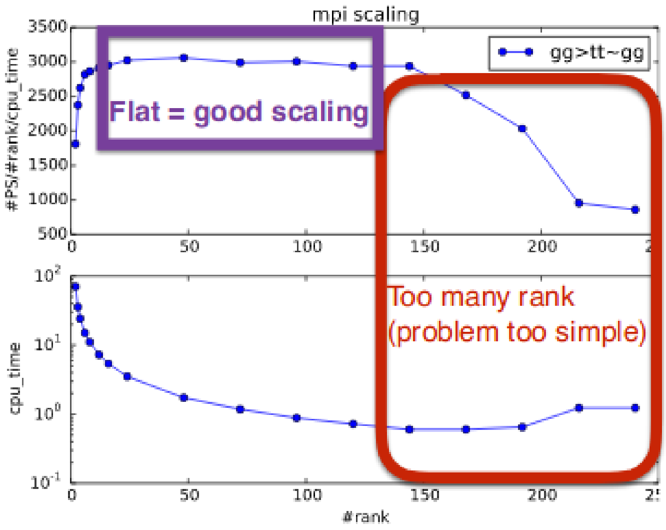
- Meanwhile, **David Lange** has built openloops with optimization (cms-sw/cmsdist#4448) and workflow 511.1 ran now within 1h15’.

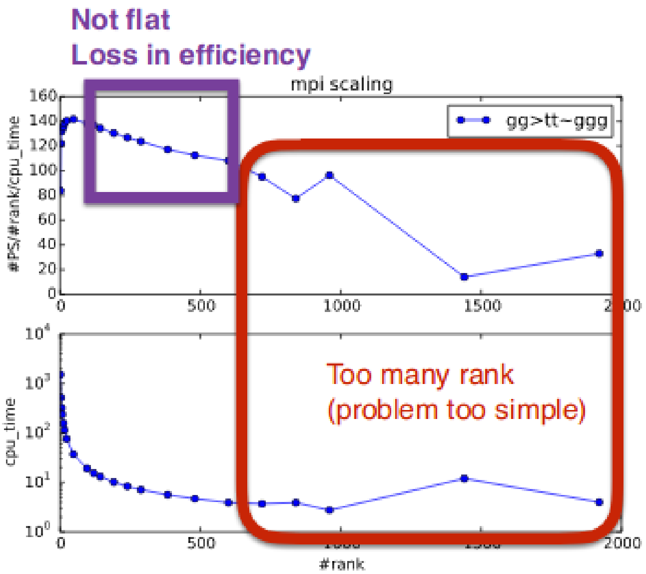
Example fix (4x faster with the CMS compilers)

```diff
- recursive function CalcFactorial(n) result(fact)
+ function CalcFactorial(n) result(fact)

    integer, intent(in) :: n
-   integer :: fact
+   integer :: fact,i

    if (n < 0) then
      write (*,*) 'factorial not defined for negative integer'
      stop
    end if

-   if (n .eq. 0) then
-     fact = 1
-   else
-     fact = n * CalcFactorial(n-1)
+   fact = 1
+   if (n .gt. 1) then
+     do i=2,n
+       fact = fact * i
+     end do
    end if
```

Generator Progress: MadGraph

- MadGraph is conservative on compiler flag option (-O1)
- Using -Ofast -> 30% faster at LO/NLO
- Timing Performance on MPI Rank numbers





Standalone MG

NLO situation is more complicated:
All phase-space point do not take the same amount of cpu time (variation by **two order** of magnitude)
Need other strategy for having the scaling

Olivier Mattelaer

Generator Progress: MadGraph

HTC vs HPC

	HTC cluster	HPC/MPI
Total waiting time		
Total cpu time		
Job granularity	faster on queue	
Infrastructure cost		+ 30% due to infiniband/OPA
GCC flag		-march=native

WINNER: The Turtle!

https://en.wikipedia.org/wiki/High-throughput_computing
The European Grid Infrastructure defines HTC as “a computing paradigm that focuses on the efficient execution of a large number of loosely-coupled tasks”, while HPC systems tend to focus on tightly coupled parallel jobs, and as such they must execute within a particular site with low-latency interconnects. Conversely, HTC systems are independent, sequential jobs that can be individually scheduled on many different computing resources across multiple administrative boundaries. HTC systems achieve this using various grid computing technologies and techniques.

Olivier Mattelaer commented: Funding from e.g. DOE has been given for HPC

Generator Progress: Sherpa

needs to be done once
results platform independent

1) **initialisation**
   - write matrix elements source code (AMEGIC)
   - write databases with list of existing processes and mapping information (AMEGIC/COMIX)

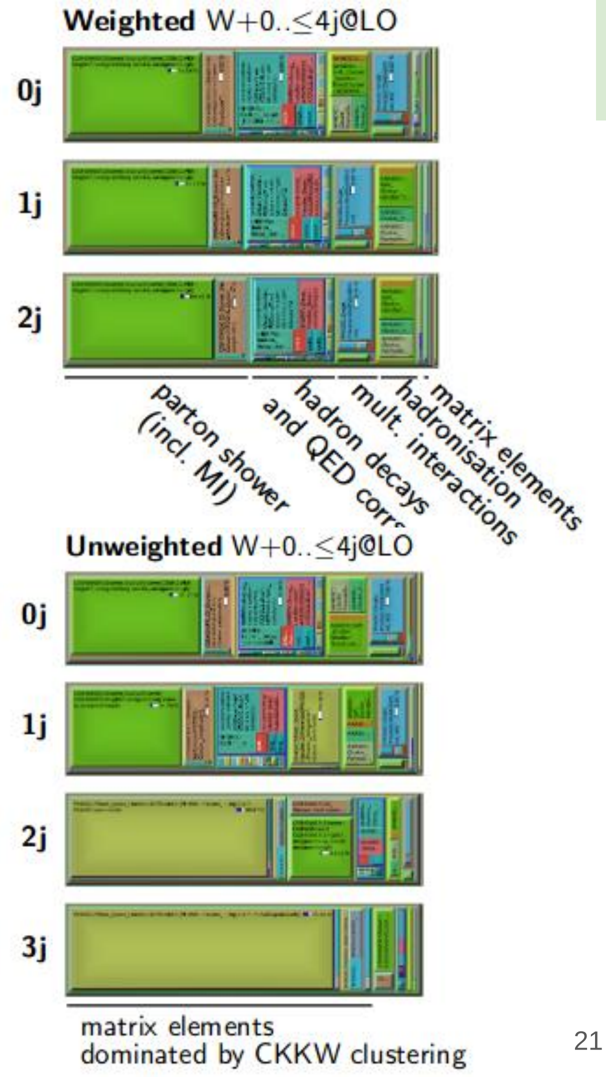
single core

2) **integration**
   - load matrix elements
   - determination of relative cross section of all subprocs
   - optimisation of phase space channels
     $\rightarrow$ make weights as uniform as possible
   - store in database

MPI parallisable
linear scaling up to few thousand cores

3) **event generation**
   - load matrix elements
   - read phase space channel parameters
   - generate events

single core
MPI parallisable but no benefits



Detailed timing break down distributions.

More improvements possible with non-default settings

approximate physics, but substantially improve performance

[Marek Schoenherr]

Generator Progress: Pythia, Herwig, etc

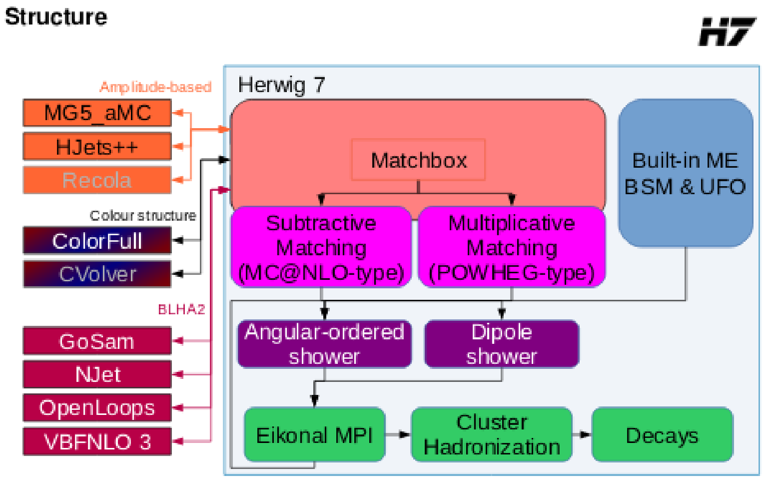
Pythia is ready for multi-threaded environment. [Philip Ilten]
Pythia8 is capable by making multiple copies but we use it with an 'external decay' package which is not thread friendly. [Christopher Jones]

Herwig7: [Peter Richardson]

Speed wasn't ignored but was clearly less important.

In Herwig the biggest speed/memory improvement would be to improve the implementation of the NLO matrix elements and phase space integration.

Many years of work for little recognition.



Fully automated, so that users can choose their process and everything is set up for them!

EvtGen: [Michal Kreps]

Main reason for multithreading is memory management

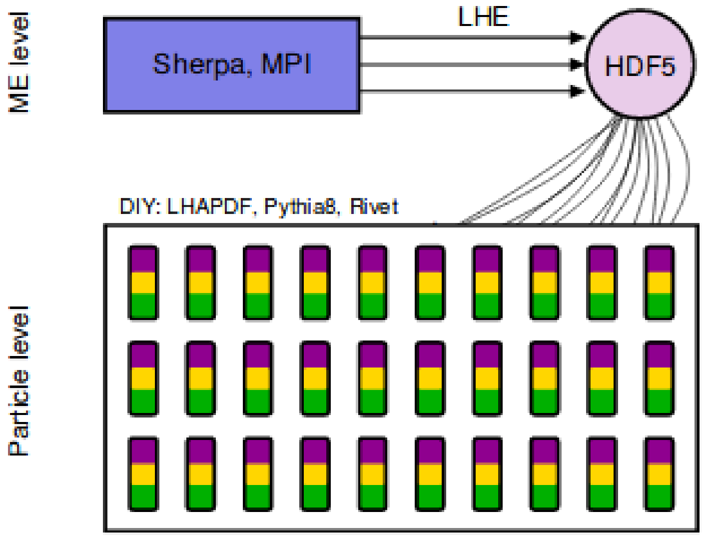
Will move to HEPMC V3 which is ready for multithreading
On scale of about half a year will try to prepare prototype of multithreading

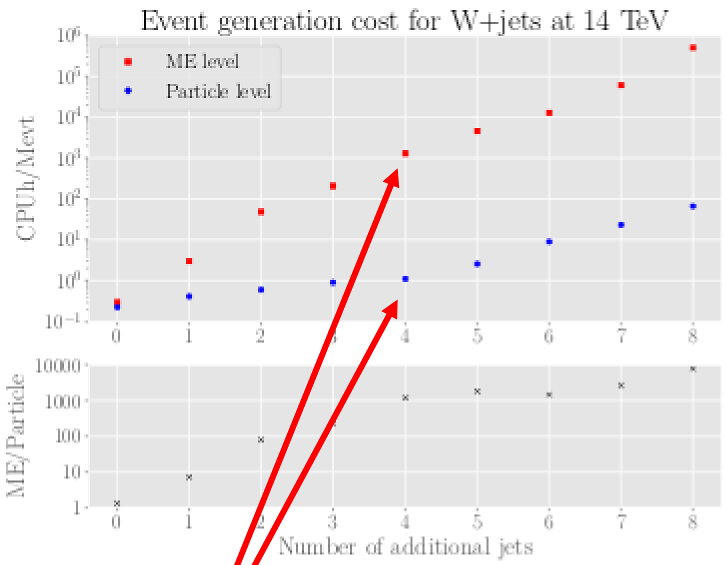
Highlight of recent effort towards HPC

Holger Schulz

High-multiplicity multi-jet merging with HPC technology W+up to 8 Jets

1. HDF5 (Hierarchical Data Format) storage for ME events: The CPU expensive part of the simulation is stored in a parton-shower independent format.
2. Particle level and merging with Pythia with ASCR's (Advanced Scientific Computing Research) DIY, which does all the low-level MPI communication. Particle level run-time up to 4 orders of magnitude faster than ME.





For comparison, CMS WJetsToLNu_TuneCUETP8M1_13TeV-madgraphMLM-pythia8 (W+0/1/2/3/4Jets) time/evnet~16s, then 1M events->4000hr (ME+PS)

Workshop Discussions

Andy Buckley, Marek Schoenherr, Stefan Hoeche

1) Reduce CPU requirements by running large scale event generation at LO:
and reweighting of LO sample to (N)NLO sample…

2) Reducing computing costs by sharing unweighted matrix-element event samples between experiments:

Format details: gridpacks? LHEs? Sherpa intermediates? HDF5? Hybrid?
Ideally factor 2 gain in computing... but merging/unweighting also slow?
Stefan Frixione is against this option as it doesn't follow the tradition that two experiments xcheck each other independently.

3) Suggestion for experiment technical RAs to make "deep" contributions in MC teams, for issues where experiments incentivised and theorists not

4) Other ways that generator codes can be better aligned to usage/needs of experiment MC production campaigns?

Workshop Discussions

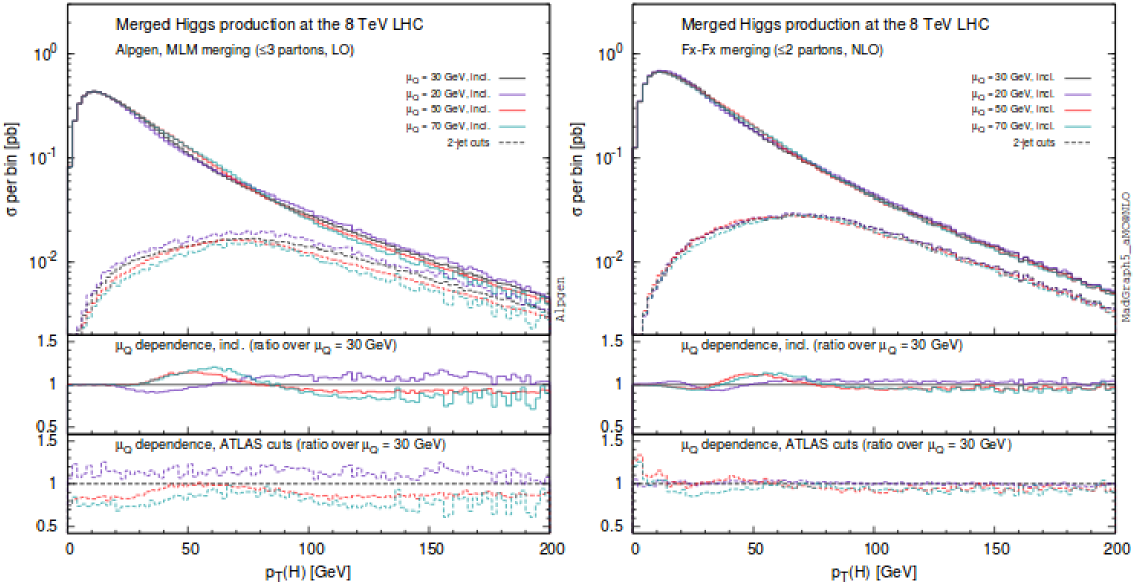
[Stefano Frixione](#)

- **Better be right than fast**
- (N)NLO simulations allow one to treat theoretical systematics seriously. This has a very direct impact on both SM and BSM physics
- There is no way to guess the local (in phase-space) behaviour of systematics without performing simulations
- One typical example: merging scale variations for a given PTJ analysis cut. NLO and LO have typically different behaviours.
- Unweighting events means spending time earlier, but saving It (plus disk space) later

Reweighting non-QCD corrections is a tricky business

Merging: LO ⟶ NLO

Left: LO (Alpgen). Right: FxFx (MadGraph5_aMC@NLO)

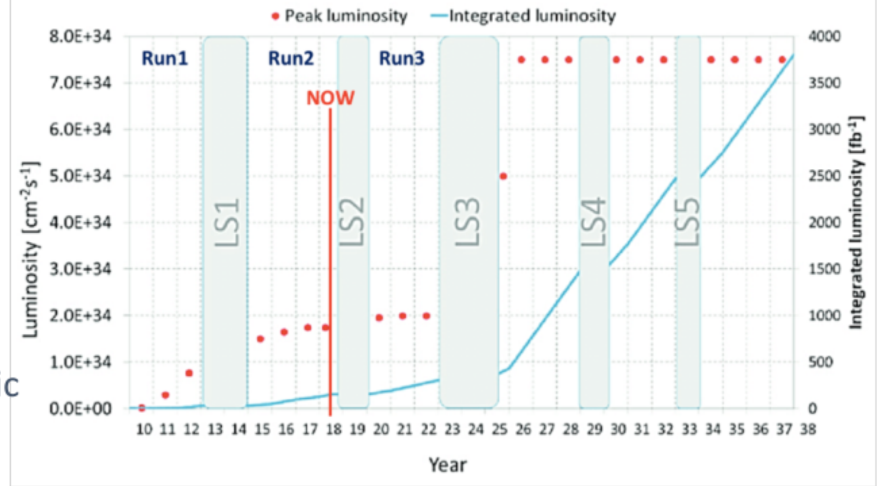


Workshop Discussions

Efe Yazgan

Beyond Run II

- Generation will only be the 3rd CPU consumer after reconstruction and detector simulation, however
- much larger samples and disk space to match data statistics
- precision measurements; top mass, W mass, weak mixing angle, …
- larger alternative samples for systematic uncertainties
- precise differential distributions and tails of the phase space regions.
- more precise calculations: NLO, NNLO, and beyond depending on the process → negative weights
- NLO QCD x EWK corrections with high multiplicity final states, for both virtual and real contributions + parton shower



→ requires much larger samples, improved PDFs, …,

→ and RIVETized (or similar) data at the extremes of the phase-space regions to improve modelling

→ *To make it technically very easy, CMS provides particle-level objects in nano-aod – and simple to produce from MiniAod*

→ *GenJets w/ hadron-flavor info*

→ *Dressed leptons*

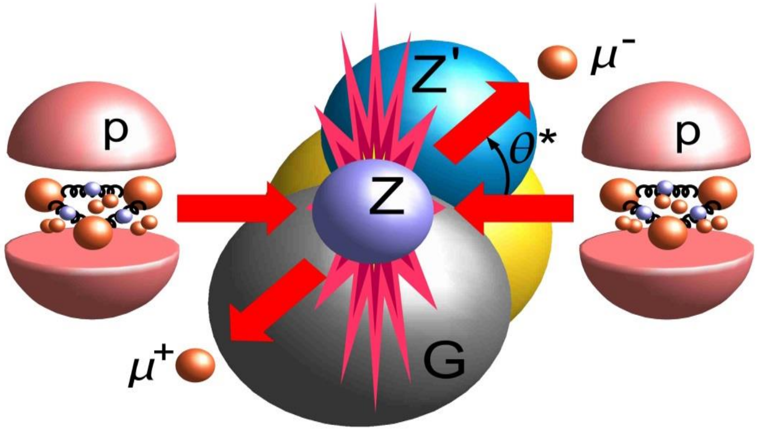
- Use common generator level events between experiments? → x~2 for free event production.
- Find a common approach for MC collaborations for the details of the implementations?
- Can physicists be supported for MC (support) positions?

Summary

- Quite fruitful progress and discussions
- Start figuring out major challenges for software and computing come in the future
- To build further collaboration mechanism between different communities
- Still a long way to go!

CMS MC Simulation Overview

– **Hard process/Matrix Element generation**:
Desired process up to parton level
using perturbative QCD



– **Parton Shower/Hadronization**:
QCD and QED emissions down to a low scale,
and produces hadrons from QCD partons

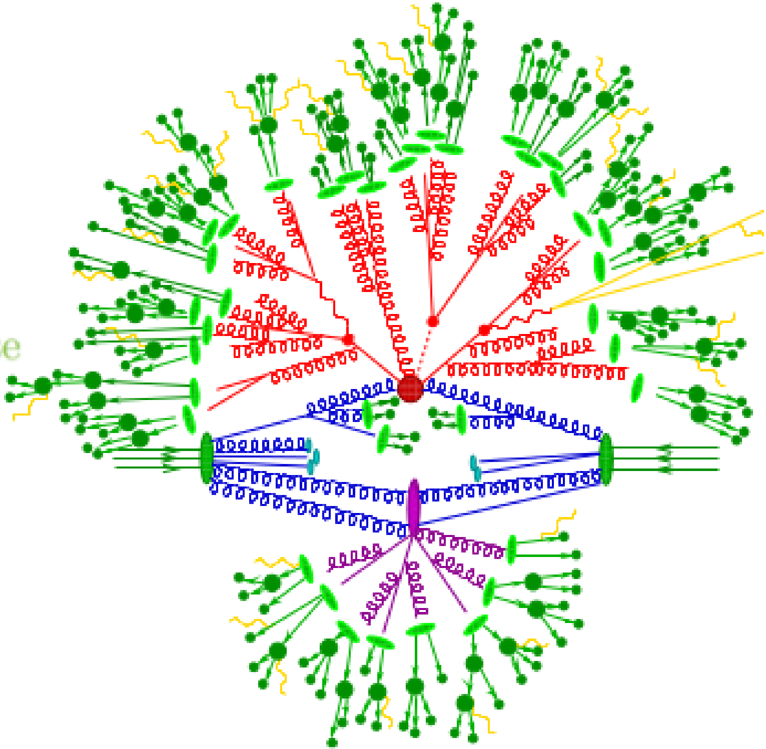
– *Multiple Parton Interaction*

– *Detector Simulation and Digitization*:
Detailed Geant4 simulation of the interactions of the outgoing particles with the CMS detector, followed by simulation of detector electronics and creation of simulated raw data

– *Reconstruction*:
Reconstruction of simulated raw data into higher level physics objects
To a good approximation, identical code as runs on real data

LHE –> GEN–> SIM–> DIGI–> RECO

MC Generator



LHC is a QCD factory:

NLO and/or multi-leg/merged-multiplicity

Factorised approach may lead to large uncertainties:

“tune” and PS weights etc needed

CMS MC Workflow

Basic paradigm for event generation

– C++ module making calls to linked external generator code to produce HepMC::GenEvent to be stored as EDM

Matrix element generators which generate LHE files (Madgraph, POWHEG, ...)

– Loosely coupled to CMSSW, calling of external generation script handled by an integrated CMSSW module **“externalLHEProducer”**

ascii LHE files are transient and immediately packed into binary compressed format

Gridpack Production

Gridpack preparation:

LSF or Condor or CMS-Connect (grid-like condor jobs using CMS Global Pool)

Gridpack size can be an issue (>500MB for the tarball and 5GB decompressed)

We maintain scripts for all the major generators to produce gridpack tarballs

– Madgraph aMC@NLO

tt012FxFx ~72h@lxplus batch

DY01234MLM ~15h@cms-connect

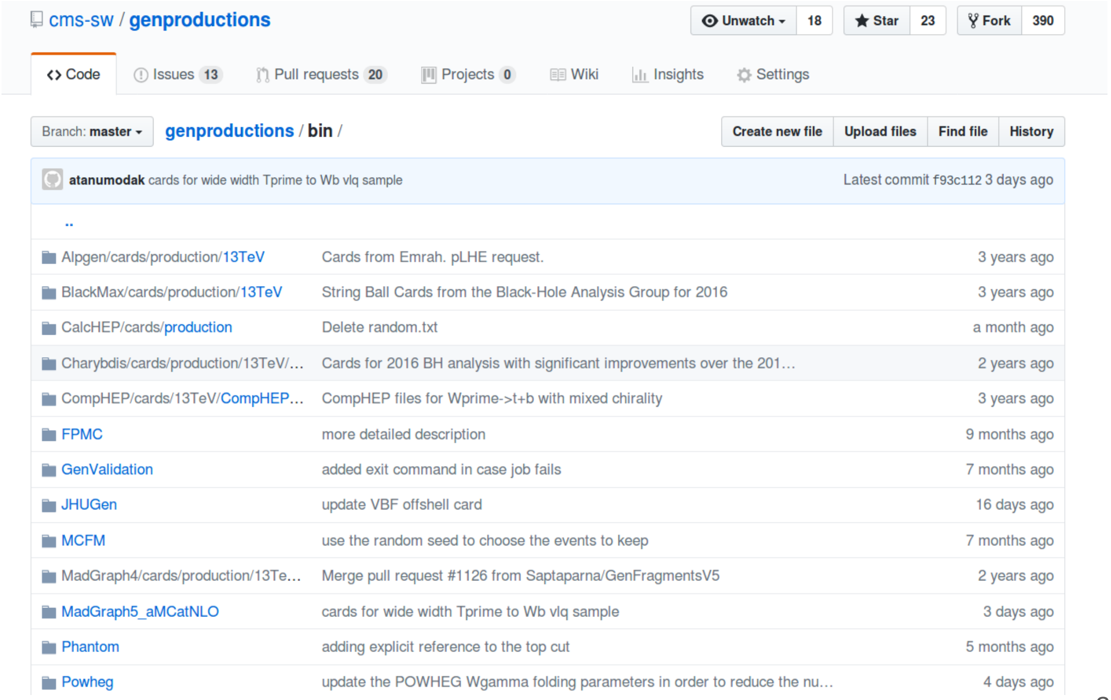
– POWHEG

– Sherpa, Herwig7 and others

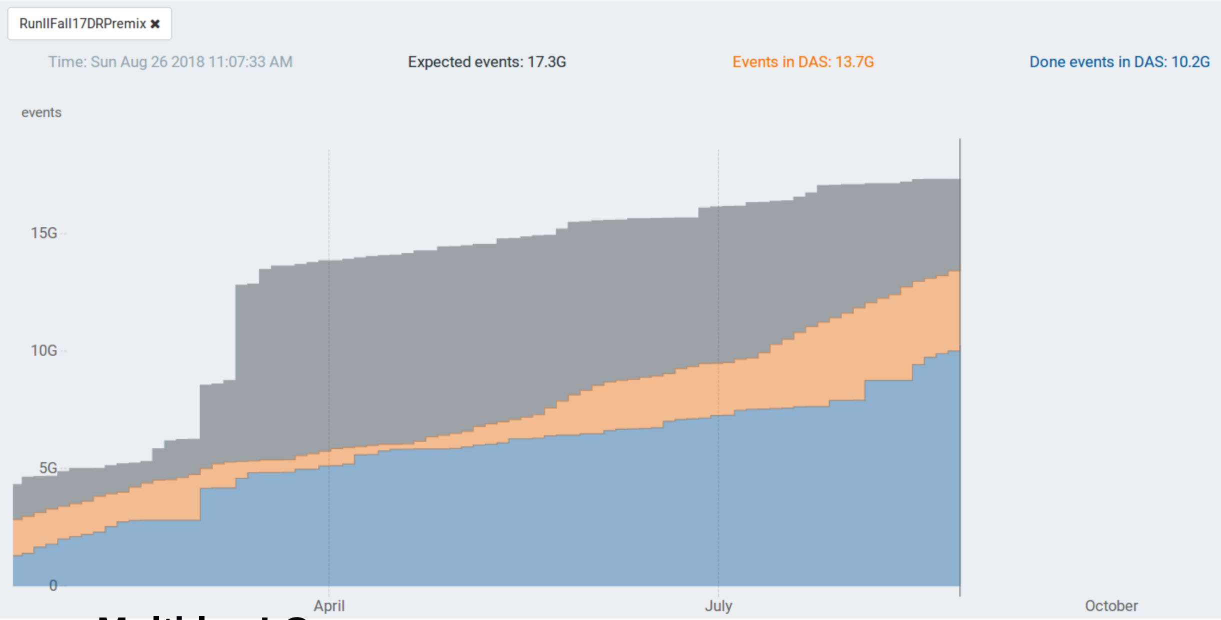
tt0123 MEPS Sherpa

~70h@cms-connect

~O(100s)/event



CMS MC Production Status



Multi-leg LO

- up to ~10s/gen-evt
- ~50% matching efficiency -> 20s/full- sim-evt

Multi-leg NLO

- up to ~30s/gen-evt
- ~30% matching efficiency -> 100s/full- sim-evt

Taking 2017 as an example:

15 B (+ some other production ~ 20 B) in 8 months

- **GEN-SIM-DIGI-RECO** ~85 sec/evt
- **60k cores** (~1/3 of the CMS production power)

- Large fraction of negative weights of up to ~40% -> larger samples!

MC requirements: High Performance Computing

Talk at CMS Fall18 CMS Offline&Computing Week

Generators

No generators can be run concurrently

Pythia8 is capable by making multiple copies but we use it with an 'external decay' package which is not thread friendly

High filtering jobs make for low efficiency

We spend a high fraction of the job time in the generator, not Geant4

Gets worse as we add more threads

Gridpacks

Run via a process external to cmsRun

Spend large percentage of time untarring files

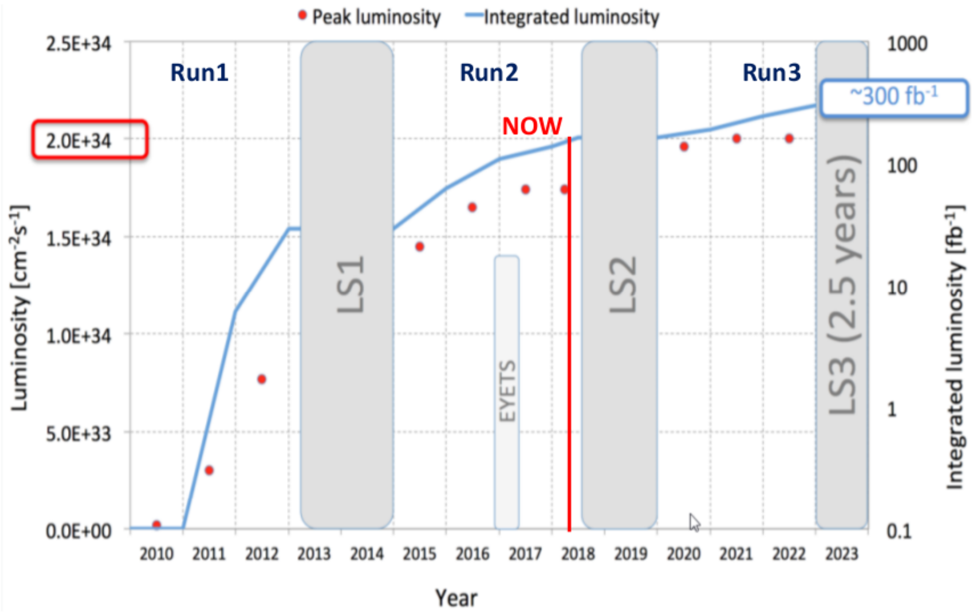
very low CPU efficiency

Primarily only use 1 thread

Heterogeneous Architectures and HPC Efficiency

Dr Christopher Jones *FNAL*

Summary



Need to "fight" against conflicting requirements:

– (Much) larger datasets
– Increased measurement precision
– Need for alternative samples for systematics
– Flattening of computing resources (both cpu and disk space)