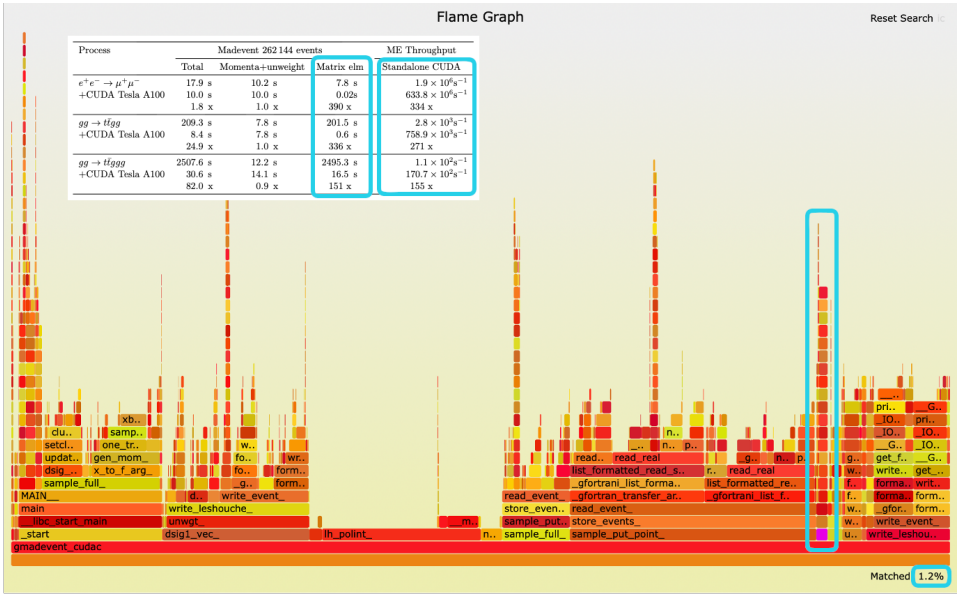

CERN

Acceleration beyond lowest order event generation

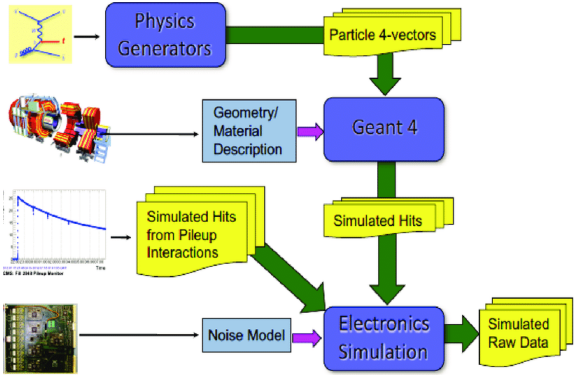
Parallelisation of event reweighing routines within the MG5aMC framework

Zenny Wettersten (CERN)
Olivier Mattelaer (UCLouvain)
Stefan Roiser (CERN)
Robert Schöfbeck (HEPHY)
Andrea Valassi (CERN)


Flame Graph
Reset Search
Process
Madevent 262 144 events
Total
Momenta+unweight
Matrix elm
ME Throughput
Standalone CUDA
$e^+e^- \to \mu^+\mu^-$
17.9 s
10.2 s
7.8 s
$1.9 \times 10^6 s^{-1}$
+CUDA Tesla A100
10.0 s
10.0 s
0.02s
$633.8 \times 10^6 s^{-1}$
1.8 x
1.0 x
390 x
334 x
$gg \to t\bar{t}gg$
209.3 s
7.8 s
201.5 s
$2.8 \times 10^3 s^{-1}$
+CUDA Tesla A100
8.4 s
7.8 s
0.6 s
$758.9 \times 10^3 s^{-1}$
24.9 x
1.0 x
336 x
271 x
$gg \to t\bar{t}ggg$
2507.6 s
12.2 s
2495.3 s
$1.1 \times 10^2 s^{-1}$
+CUDA Tesla A100
30.6 s
14.1 s
16.5 s
$170.7 \times 10^2 s^{-1}$
82.0 x
0.9 x
151 x
155 x
MAIN__
main
__libc_start_main
_start
gmadevent_cudac
sample_full_
dsig_...
updat..
setcl..
clu..
xb..
dsig1_vec_
write_event_
unwgt_
write_leshouche_
lh_polint_
sample_full_
sample_put_point_
store_events_
read_event_
read_real
Matched 1.2%

What if we could forego the overhead?

Reweighing



CMS simulation workflow.
(Hildreth; Ivanchenko; Lange, 2017.).

Simulation stages factorise, i.e simulated observable F,

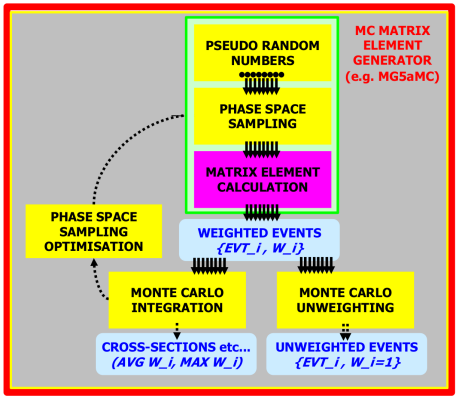
$$F(\chi, \upsilon) = X(\chi)Y(\upsilon).$$

If $\chi \to \chi'$ only changes X,

$$F(\chi', \upsilon) = X(\chi')Y(\upsilon) = X(\chi')Y(\upsilon)\left(X(\chi)/X(\chi)\right) = F(\chi, \upsilon)\left(X(\chi')/X(\chi)\right).$$

Reweighing

- If new physics doesn't affect later simulation stages:
  Only need to regenerate events
  (Note: **Very** *hard condition)*
- $\rightarrow$ Can recycle simulations
  (Note: Must simulate orig. evt.)
- Furthermore: Event generation factorises, too



Monte Carlo integration

- Solve integrals stochastically (stochastic Riemann integration)
- Given integrand $F(x)$, sample x-space and evaluate F
- Probe x-space until relevant region has been studied sufficiently
- Assign each x_i a weight W_i based on its relevance
- Evaluate integral as

$$\int F(x)\, dx \simeq \sum_i W_i \cdot F(x_i).$$

Event weights

- For MC generation, event weights are (Mattelaer, 2016.)

$$W = \underbrace{|M(p)|^2}_{\substack{\text{depends on} \\ \text{new physics}}} \times \underbrace{f_1(x_1, \mu_F) \times f_2(x_2, \mu_F) \times \Omega_{PS}}_{\text{does not depend on new physics}}.$$

- The weight W' given new physics is then

$$W' = W' \times \left(\frac{|M(p)|^2}{|M(p)|^2} \right) = W \times \left(\frac{|M'(p)|^2}{|M(p)|^2} \right).$$

- $\rightarrow$ Only need to reevaluate the matrix element

Reweighing, technically

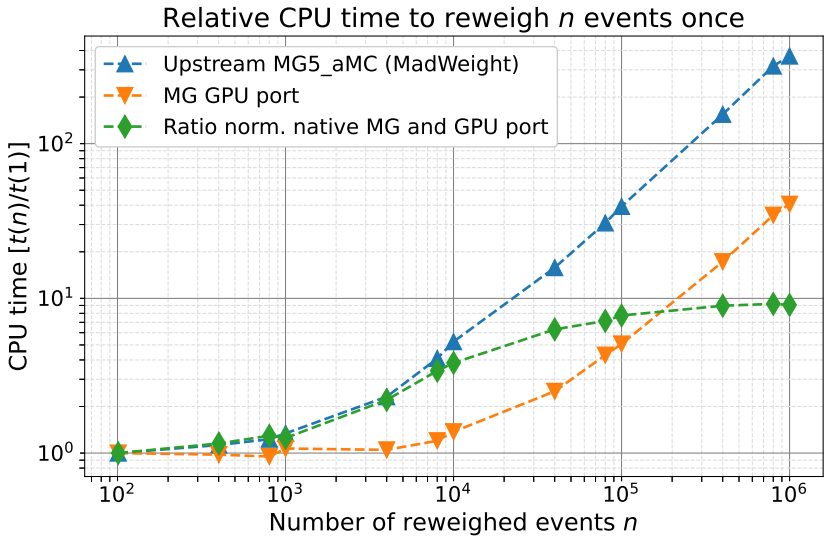
1. Take a set of generated events (stored in LHE format file)
2. Take additionally the original and the new physics parameter sets (stored in SLHA format blocks)
3. Generate routines for evaluating MEs
4. Evaluate MEs with new physics but original kinematics
5. Evaluate and output the new weight from ratio above

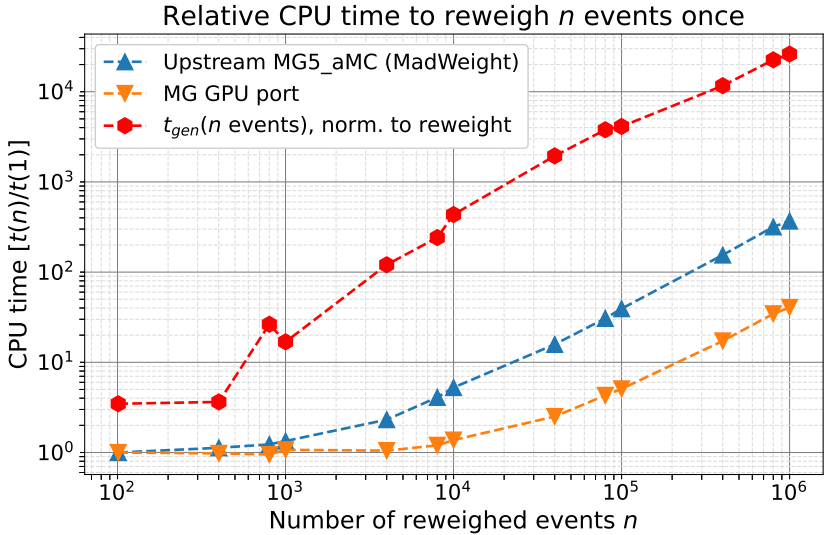
Introducing TEAWREX

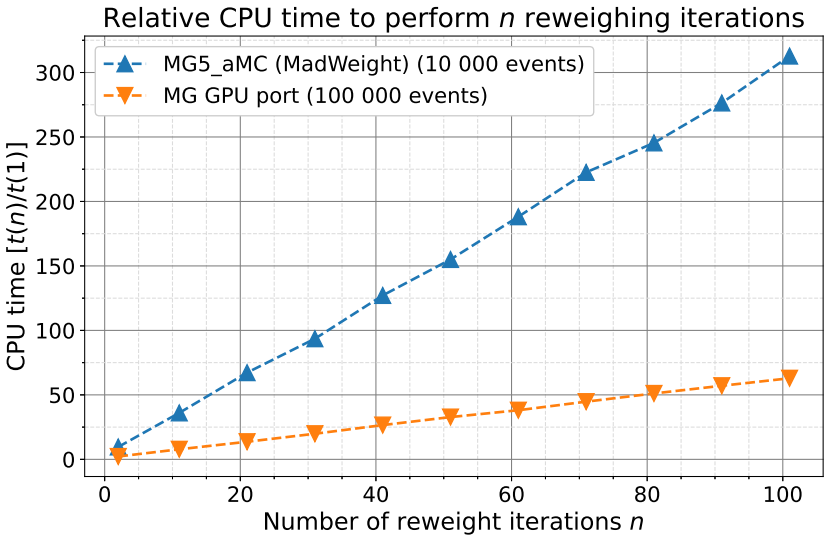
- C++ library for performing event reweighing
- Event generator agnostic (LHE file format standard)
- ME evaluation tool agnostic (currently implemented with MG)
- Assumes LHE file format and SLHA standard parameter cards
- Takes ME eval. function as input to run reweighing scheme

MG5aMC GPU port implementation

1. Port allows ME evaluation on GPUs and vector CPUs
2. Build minimal C++ program calling this ME eval. function
3. Pass event data parsed from LHE file to eval. function
4. Overwrite SLHA parameter card
5. Run eval. function for new parameters
6. Iterate over steps 3-5 for all considered parameter sets


Relative CPU time to reweigh n events once
Upstream MG5_aMC (MadWeight)
MG GPU port
Ratio norm. native MG and GPU port
CPU time $[t(n)/t(1)]$
10^0
10^1
10^2
10^2
10^3
10^4
10^5
10^6
Number of reweighed events n


Relative CPU time to reweigh n events once
Upstream MG5_aMC (MadWeight)
MG GPU port
t_{gen}(n events), norm. to reweight
CPU time [$t(n)/t(1)$]
10^0
10^1
10^2
10^3
10^4
10^2
10^3
10^4
10^5
10^6
Number of reweighed events n


Relative CPU time to perform n reweighing iterations
MG5_aMC (MadWeight) (10 000 events)
MG GPU port (100 000 events)
CPU time [$t(n)/t(1)$]
Number of reweight iterations n
0
50
100
150
200
250
300
0
20
40
60
80
100

Status and outlook

- Current situation:
  - Proof of concept sent to CMS for testing
  - Initial results suggest acceleration on par with standalone MEs compared to full event generation in MG GPU port
- Future possibilities:
  - Incorporate TEAWREX and MG code generation
  - Differential studies in HEP parameters, see (Valassi, 2020.)
  - More extensive reweighing process detail options (specific vs. summed helicities, colour flows etc.)

Overarching outlook

- Accelerated reweighing allows for extensive physics probing
- Could make differentiable studies in HEP parameters accessible
- Upstream MG5aMC can reweigh at NLO accuracy
  - We are looking into NLO event generation porting
  - FKS subtraction splits NLO contributions into three parts, two of which can be evaluated using LO machinery
  - $\rightarrow$ More heterogeneous computing?
- Parallelism can bring acceleration beyond LO event generation


CERN

Backup slides

Hardware statistics

Measurements of proc. $e^+e^- \to 5\gamma$.

- CPU: Intel Core i7-1165G7
  - Consumer grade CPU
  - 4 physical cores, 8 threads
  - Clock speed 2.80 GHz
- GPU: NVidia A100 40GB
  - HPC grade GPU
  - FP64 performance: 9.7 TFLOPS

Measurements are of total CPU time.

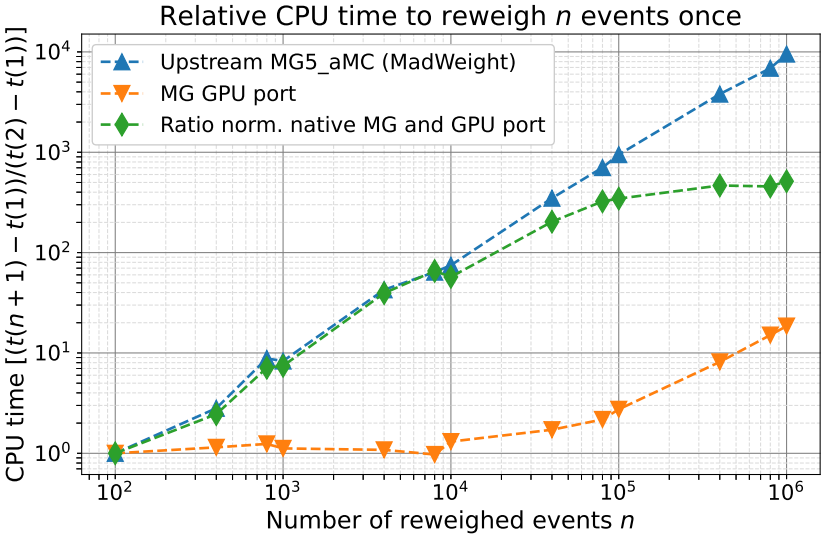
GA100 SM. (Nvidia, 2020.)

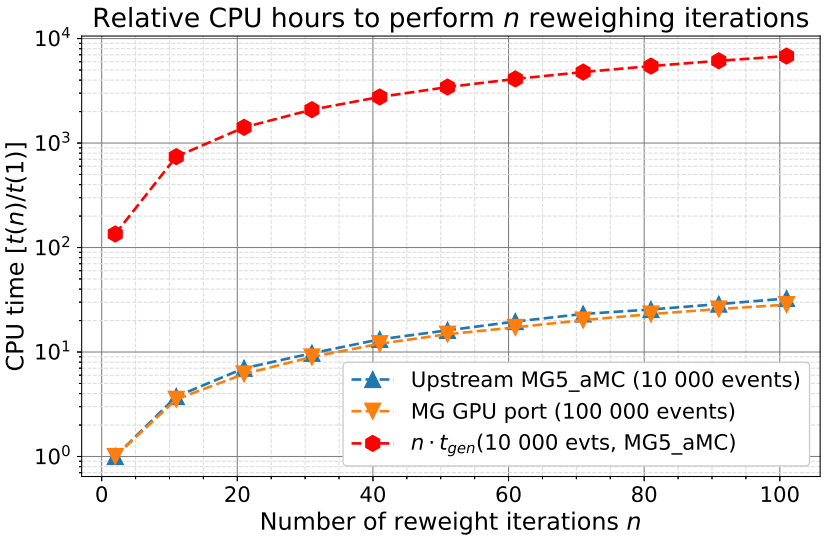
Unweighing and reweighing

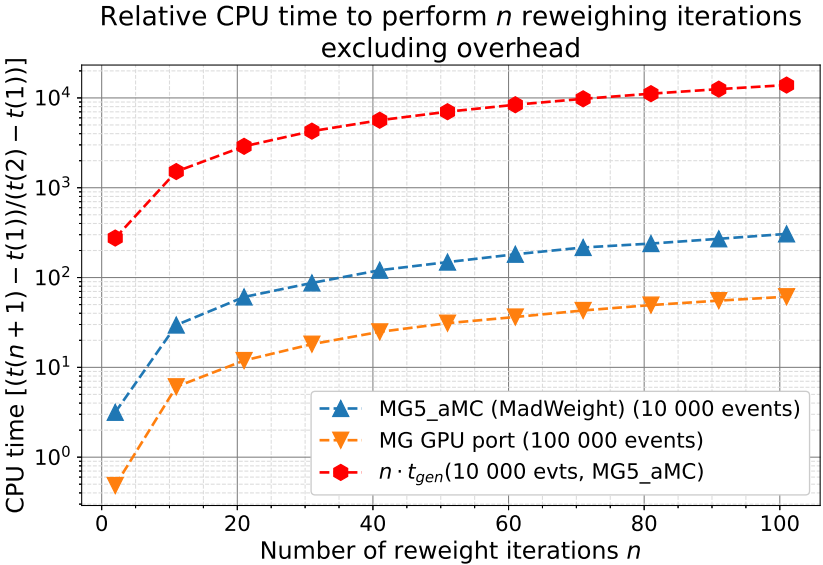
- For evt. gen., typically want each event to be equally relevant
- Unweighing: Sample phase space s.t. events have same weight
- Still stochastic integral:

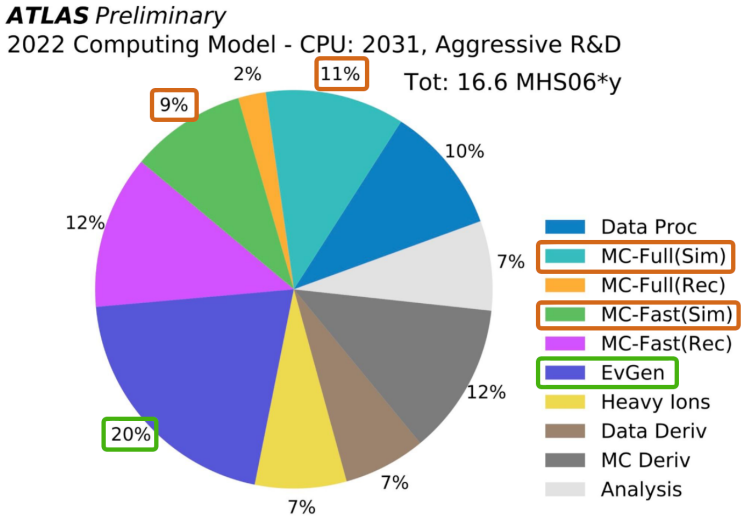
$$\int F(x)\, dx \simeq \sum_i W_i \cdot F(x_i),$$

but now $W_i = W_j$ for all i, j.


Relative CPU time to reweigh n events once
Upstream MG5_aMC (MadWeight)
MG GPU port
Ratio norm. native MG and GPU port
CPU time $[(t(n+1)-t(1))/(t(2)-t(1))]$
Number of reweighed events n
10^0
10^1
10^2
10^3
10^4
10^2
10^3
10^4
10^5
10^6


Relative CPU hours to perform n reweighing iterations
CPU time [$t(n)/t(1)$]
10^0
10^1
10^2
10^3
10^4
0
20
40
60
80
100
Number of reweight iterations n
Upstream MG5_aMC (10 000 events)
MG GPU port (100 000 events)
$n \cdot t_{gen}$(10 000 evts, MG5_aMC)


Relative CPU time to perform n reweighing iterations excluding overhead
CPU time $[(t(n+1)-t(1))/(t(2)-t(1))]$
10^0
10^1
10^2
10^3
10^4
0
20
40
60
80
100
Number of reweight iterations n
MG5_aMC (MadWeight) (10 000 events)
MG GPU port (100 000 events)
$n \cdot t_{gen}$(10 000 evts, MG5_aMC)



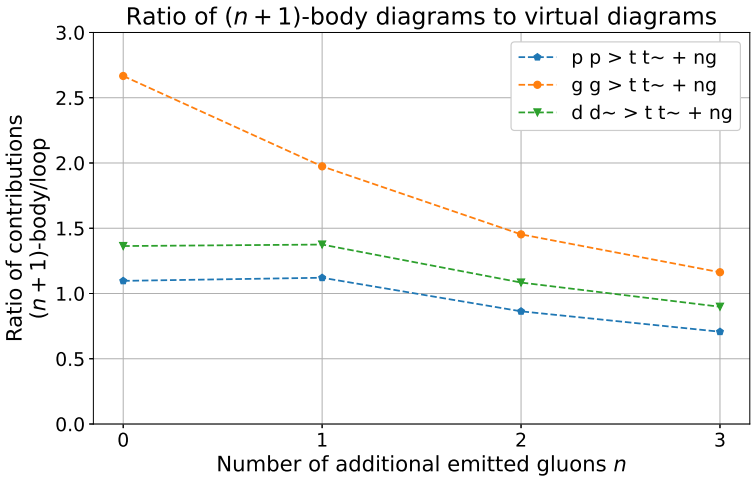
Atlas proj. CPU usage 2031, aggressive R&D. (CERN-LHCC-2022-005)

NLO contributions in FKS subtraction

- Born contributions: LO tree diagrams
  Same calculations as LO
- Real contributions: $(n+1)$-body diagrams
  Tree diagrams with 1 additional external particle
- Virtual contributions: LO diagrams with closed internal legs
  Improper integrals over internal momenta
  Not yet clear how parallelisable they are

Loop complications

- Evaluate integral with Cauchy's residue theorem
- Integrand splitting method varies dep. integrand
- Sometimes needs quadruple (FP128) precision
- *pp* collisions: $\sim 40 - 50\%$ of NLO ev. gen. CPU time


Ratio of $(n+1)$-body diagrams to virtual diagrams
Ratio of contributions
$(n+1)$-body/loop
Number of additional emitted gluons n
p p > t t~ + ng
g g > t t~ + ng
d d~ > t t~ + ng
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
1
2
3