

Madevent

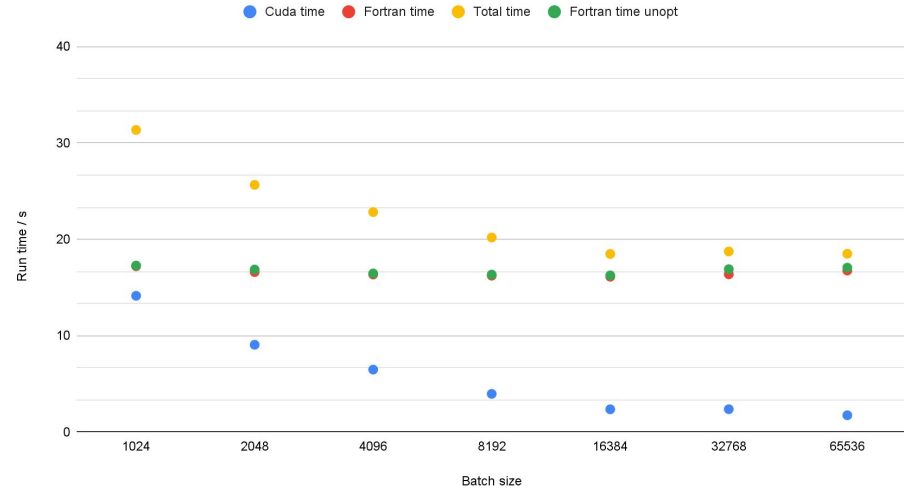
Throughput Measurements and other small things

What is the best batch size for madevent??

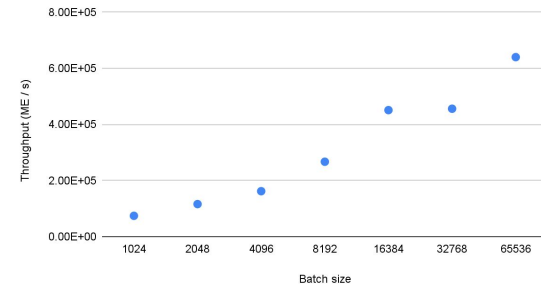
- Use madevent gg → ttgg.mad as starting point (this includes GPU-assisted unweighting)
- Run $2^{20} = 1,048,576$ events with various madevent batch sizes
- 5 runs each, TeslaA100
- awk → mean values → plot

- Finding a sweet spot at 16k - 32k events
- 64k is 0.5s faster, but not easy to achieve (see later)

Madevent: Run times vs batch size gg → ttgg



Madevent: ME throughput gg → ttgg



Config that's running

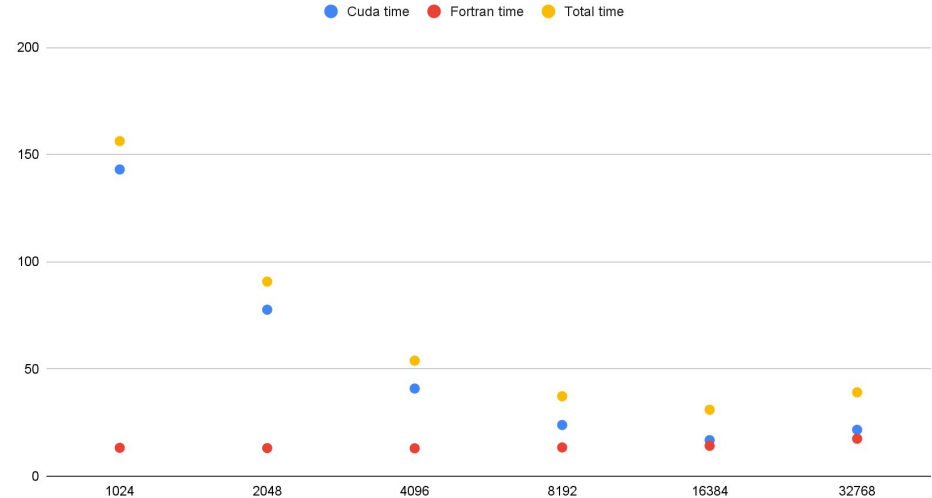
```
1 ! Fortran bridge mode (CppOnly=1, FortranOnly=0, BothQuiet=-1, BothDebug=-2)
${SIZE} ! Number of events in a single C++ or CUDA iteration (nb_page_loop)
${NUMEVENT} 1 1 ! Number of events and max and min iterations
0.000001 ! Accuracy (ignored because max iterations = min iterations)
0 ! Grid Adjustment 0=none, 2=adjust (NB if = 0, ftn26 will still be used if present)
1 ! Suppress Amplitude 1=yes (i.e. use MadEvent single-diagram enhancement)
0 ! Helicity Sum/event 0=exact
1 ! Channel number (1-N) for single-diagram enhancement multi-channel (NB used even if suppress amplitude is
0!)
```

What is the best batch size for madevent??

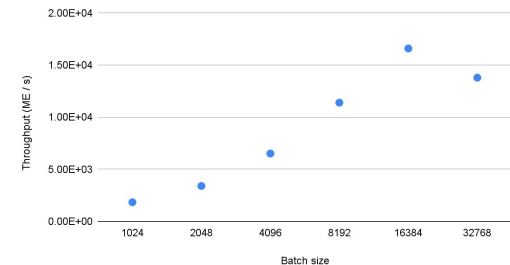
- Use madevent **gg** → **ttggg**.mad as starting point (**no GPU-assisted unweighting**)
- Run $2^{18} = 264,192$ events with various madevent batch sizes
- 5 runs each, TeslaA100

- Finding a sweet spot at 16k events
- madevent slows down at 32k events

Madevent: Run times vs batch size gg -> ttggg



Madevent: ME throughput gg -> ttggg



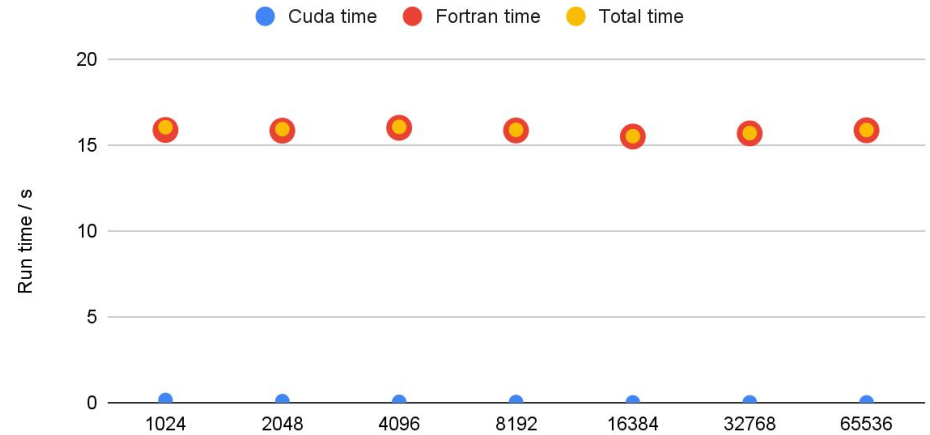
What is the best batch size for madevent??

- Use madevent **ee** → **mumu**.mad as starting point (**no GPU-assisted unweighting**)
- Run $2^{20} = 1,048,576$ events with various madevent batch sizes
- 5 runs each, TeslaA100

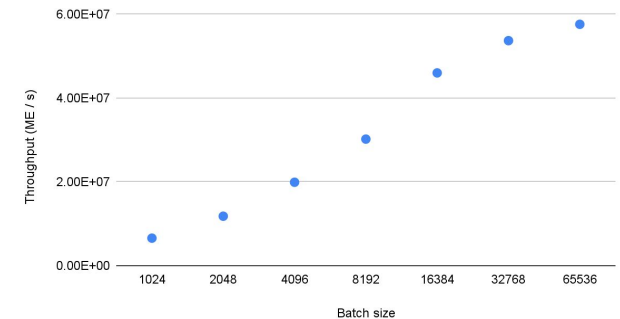
- Larger is better for GPU
- Fortran time almost completely unaffected

All in all: 16k seems to be the best

Run times ee --> mumu



Madevent: me throughput ee -> mumu

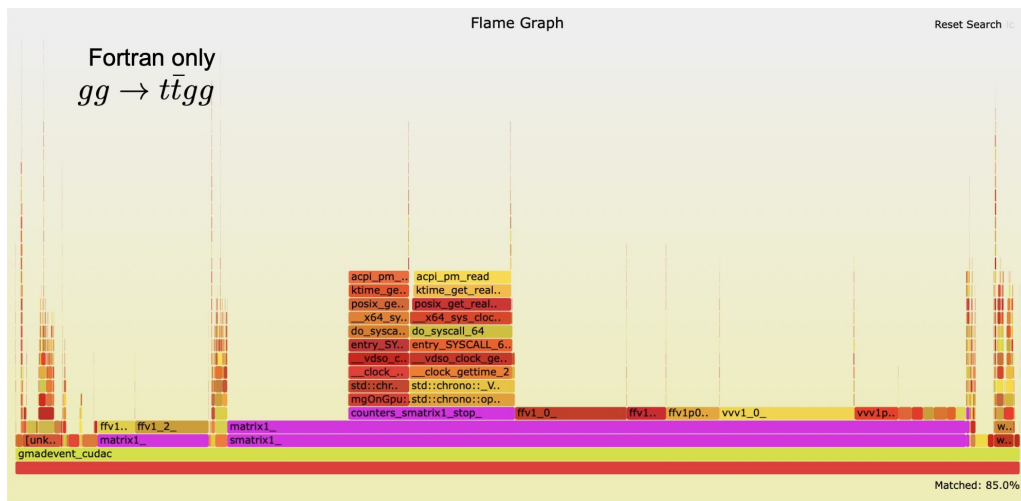


Running >16k batches in madevent?

- Lugano Hackathon:
 - Went from 8k (?) batches to 16k
 - Hit a hard limit after that (crash in fortran)
- For today's study, I wanted 64k ...
 - Recompiled everything with -g -O0; also fortran
NB: `make GLOBAL_FLAG="-g --fbounds-check" OPTFLAGS="-g"`
 - Crash happened at the beginning of a function call in fortran
→ stack overflow
- Increased stack size from 8 → 32 Mb (`ulimit -s`) fixed the crash
- **Therefore:** two ways to increase batch size in madevent (if needed at any point):
 - Trace down stack arrays of the shape `arr[VECSIZE_MEMMAX]` in madevent; put on heap
 - Tell everybody to run with more stack space

A curiosity on measuring times

- Puzzled why "counters" show up in the flame graph (CHEP23 talk)
- Turns out they are compiled without -O3
- gmadevent+cuda
 - Run times reduced by 0.1s (see slide 2)
 - Irrelevant when running for 10s of seconds
- gmadevent (only fortran)
 - Adding -O3:
 - Total: 3:41s → 3:35 s
 - MEs only: 207s → 201 s
 - Usage of counters overly pessimises gmadevent fortran MEs



```
[0]shageboe@itscrd-a100:P1_epem_mupmum (maxWeightGPU)$ make
CUDACPP_BUILDDIR='.'
g++ -std=c++11 -Wall -Wshadow -Wextra -c counters.cc -o counters.o
gfortran -o madevent myamp.o genps.o unwgt.o setcuts.o get_color.o cuts.o cluster.o reweight.o initcluster.o addmothers.o
el -lgeneric -lpdf -lcerlib -lbias -fopenmp counters.o ompnumthreads.o -lstdc++ -Wl,--no-relax
gfortran -o ./cmadevent_cudacpp myamp.o genps.o unwgt.o setcuts.o get_color.o cuts.o cluster.o reweight.o initcluster.o
b/ -ldhelas -ldsampler -lmodel -lgeneric -lpdf -lcerlib -lbias -fopenmp counters.o ompnumthreads.o -L../lib/ -lm
if [ -f ../lib/./libmg5amc_epem_mupmum_cuda.* ]; then gfortran -o ./gmadevent_cudacpp myamp.o genps.o unwgt.o setcut
o auto_dsig1_cudacpp.o auto_dsig.o matrix1.o -L../lib/ -ldhelas -ldsampler -lmodel -lgeneric -lpdf -lcerlib -lbias
/lib/' -lstdc++ -Wl,--no-relax ; fi
```

Summary

- The sweet spot for batch size in madevent seems to be 16k at the moment
- Crashes at batches $> 16k$ are understood and fixable
- In gmadevent(fortran), we call the timers too often
- **TODO**: upstream ...