STATUS OF CMS-MG4GPU INTEGRATION

JIN CHOI

Nov. 5, 2024

For CMS-Madgraph5 Joint Meeting

INTRODUCTION

Planning for the next iteration

- ✓ Final Goal for CMS - to make sure no physics difference
  - In the current validation iteration, with LO, testing DY / W / TT, both inclusive and jet-binned
- ✓ No severe cross-section / kinematic distribution difference between v29x and v35x for LO!
- ✓ Comparison between v35x and v360 would be a good starting point
- ✓ Before starting the physics validation, **trying to make sure every tools working okay**

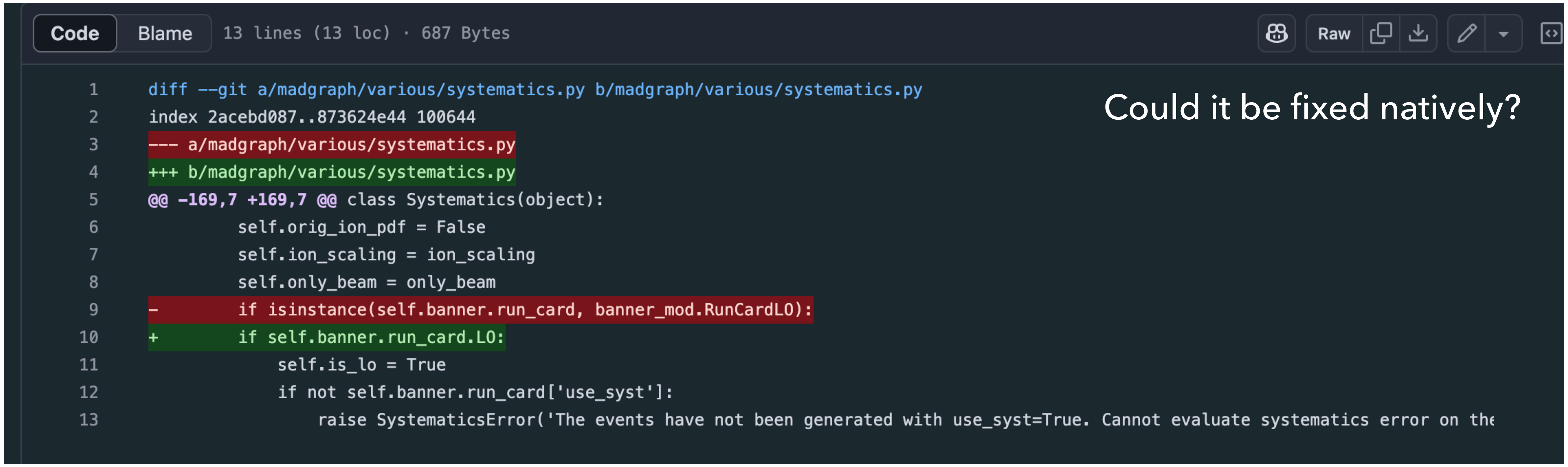
Today's focus

INTEGRATION

Patching cudacpp_for3.6.0_v1.00.00 to genproduction

✓ [Temporary Document] for instructions

✓ use_syst = True -> not working, using [0077-fix_systematics_for_simd.patch] to resolve



```
diff --git a/madgraph/various/systematics.py b/madgraph/various/systematics.py
index 2acebd087..873624e44 100644
--- a/madgraph/various/systematics.py
+++ b/madgraph/various/systematics.py
@@ -169,7 +169,7 @@ class Systematics(object):
         self.orig_ion_pdf = False
         self.ion_scaling = ion_scaling
         self.only_beam = only_beam
-        if isinstance(self.banner.run_card, banner_mod.RunCardLO):
+        if self.banner.run_card.LO:
             self.is_lo = True
             if not self.banner.run_card['use_syst']:
                 raise SystematicsError('The events have not been generated with use_syst=True. Cannot evaluate systematics error on the
```

Could it be fixed natively?

TOOLS

Migrating experiments to NERSC

- ✓ Using SLURM Batch system - easy to get high-performance isolated also interactive nodes
- ✓ Nodes configured with [64x4 AMD CPUs] or [64x2 AMD CPUs + 4 A100 GPUs]
- ✓ More sophisticated "time" output!
- ✓ Base OS is OpenSUSE - Need to use docker image to run CMSSW

Time command

- ✓ In NERSC, showing more sophisticated output than other machines, e.g.
  152.93user : CPU in user mode
  21.83system : CPU utilized by system kernel
  5:55.56elapsed : Total time
  49%CPU : CPU Utilization percentage
  (0avgtext+0avgdata 2576940maxresident)k : Memory usage
  100018inputs+2156200outputs : I/O statistics
  (1538major+1027411minor)pagefaults
  0swaps

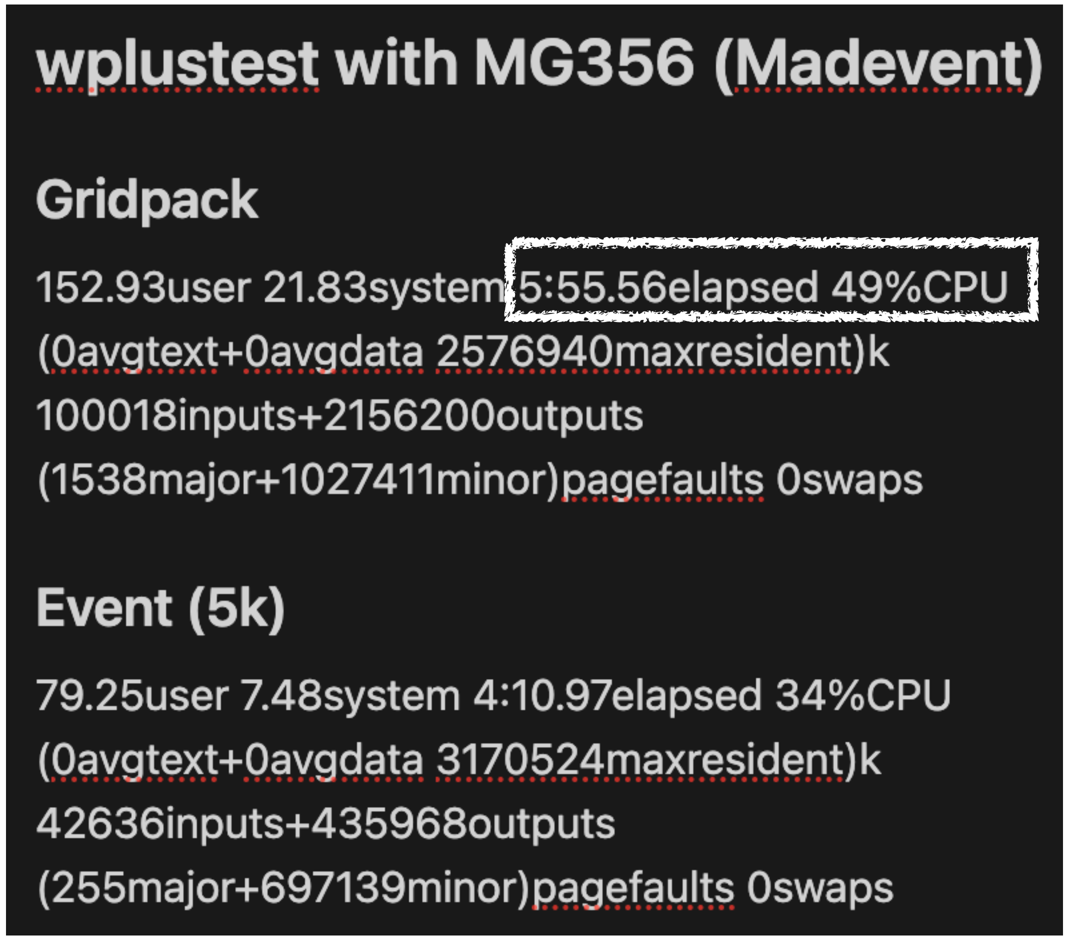
AdaptivePerf [github]

- ✓ Possible to profile with "sudo" privilege🥲 - docker images provided, but base OS is Gentoo
- ✓ Trying to figure out if it is possible to install it in el8 / el9 based docker image and profile

TESTS

MG 356 vs. MG360, madevent plugin

- Simple test with W+jets [datacards]



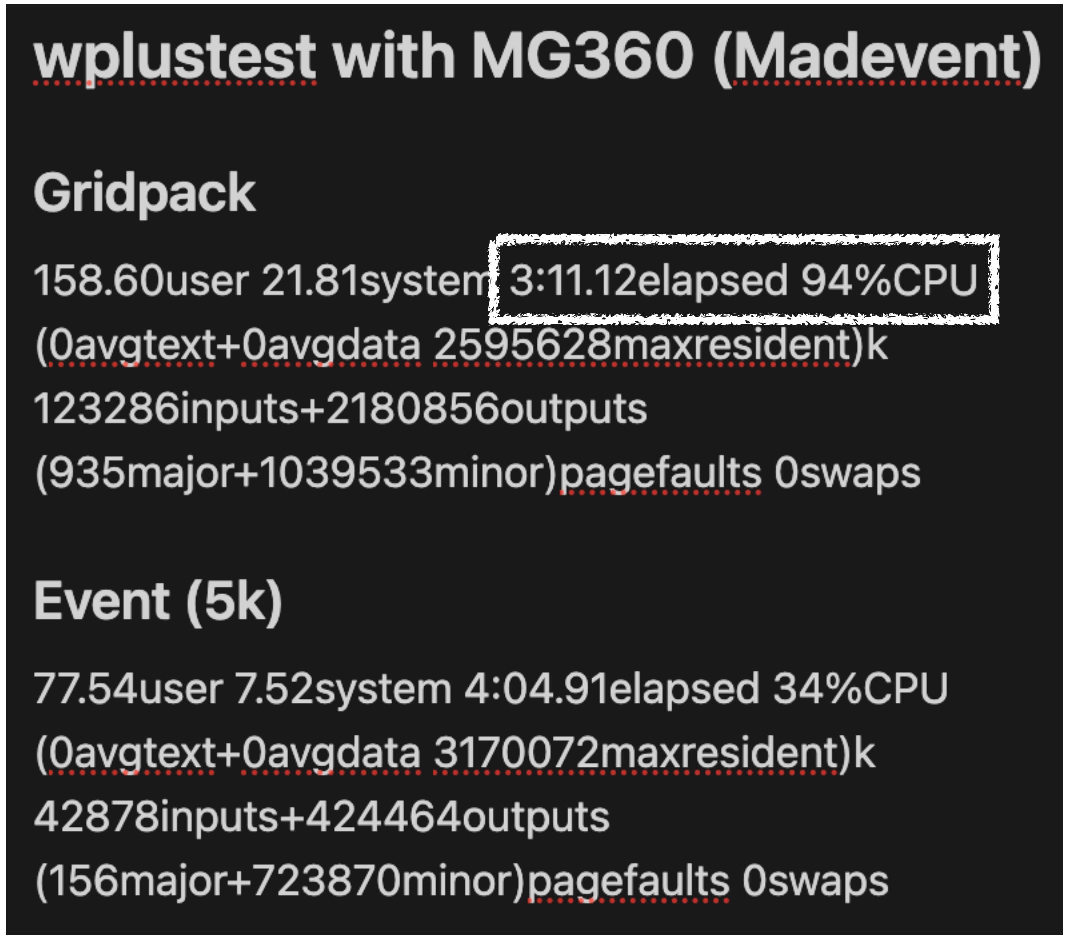
wplustest with MG356 (Madevent)

Gridpack

```
152.93user 21.83system 5:55.56elapsed 49%CPU
(0avgtext+0avgdata 2576940maxresident)k
100018inputs+2156200outputs
(1538major+1027411minor)pagefaults 0swaps
```

Event (5k)

```
79.25user 7.48system 4:10.97elapsed 34%CPU
(0avgtext+0avgdata 3170524maxresident)k
42636inputs+435968outputs
(255major+697139minor)pagefaults 0swaps
```



wplustest with MG360 (Madevent)

Gridpack

```
158.60user 21.81system 3:11.12elapsed 94%CPU
(0avgtext+0avgdata 2595628maxresident)k
123286inputs+2180856outputs
(935major+1039533minor)pagefaults 0swaps
```

Event (5k)

```
77.54user 7.52system 4:04.91elapsed 34%CPU
(0avgtext+0avgdata 3170072maxresident)k
42878inputs+424464outputs
(156major+723870minor)pagefaults 0swaps
```

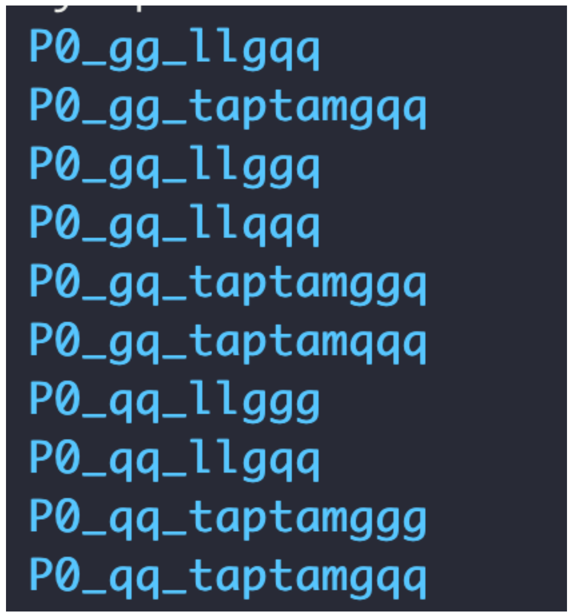
- More efficiently using CPUs, some templates changed to improve CPU efficiency (e.g. parallelized restore_data), any other parts?
- Not really need to compare speed between MG360 and previous MG's
(might relevant with MG2.9.18? - reported twice longer time for produce gridpacks in MG3xx)

TESTS

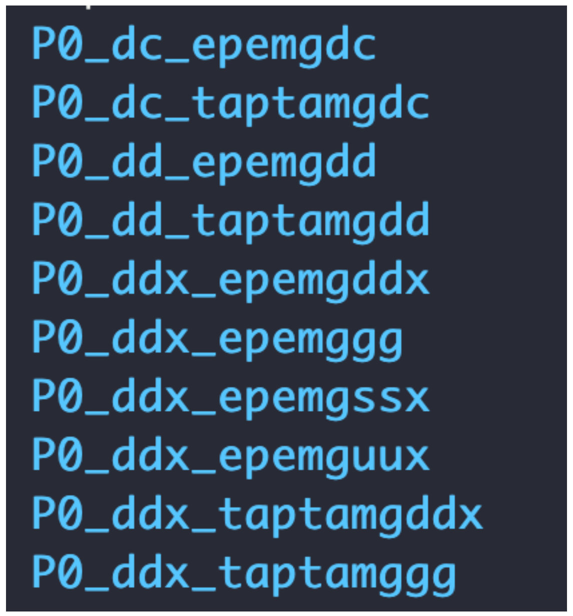
madevent vs. madevent_simd w/ fortran backend

- DY+012j / DY+3j
- Different no. of subprocess directories:

	DY+012j	DY+3j
madevent	14	10
madevent_simd	76	84



```
P0_gg_llgqq
P0_gg_taptamgqq
P0_gq_llggq
P0_gq_llqqq
P0_gq_taptamggq
P0_gq_taptamqqq
P0_qq_llggg
P0_qq_llgqq
P0_qq_taptamggg
P0_qq_taptamgqq
```

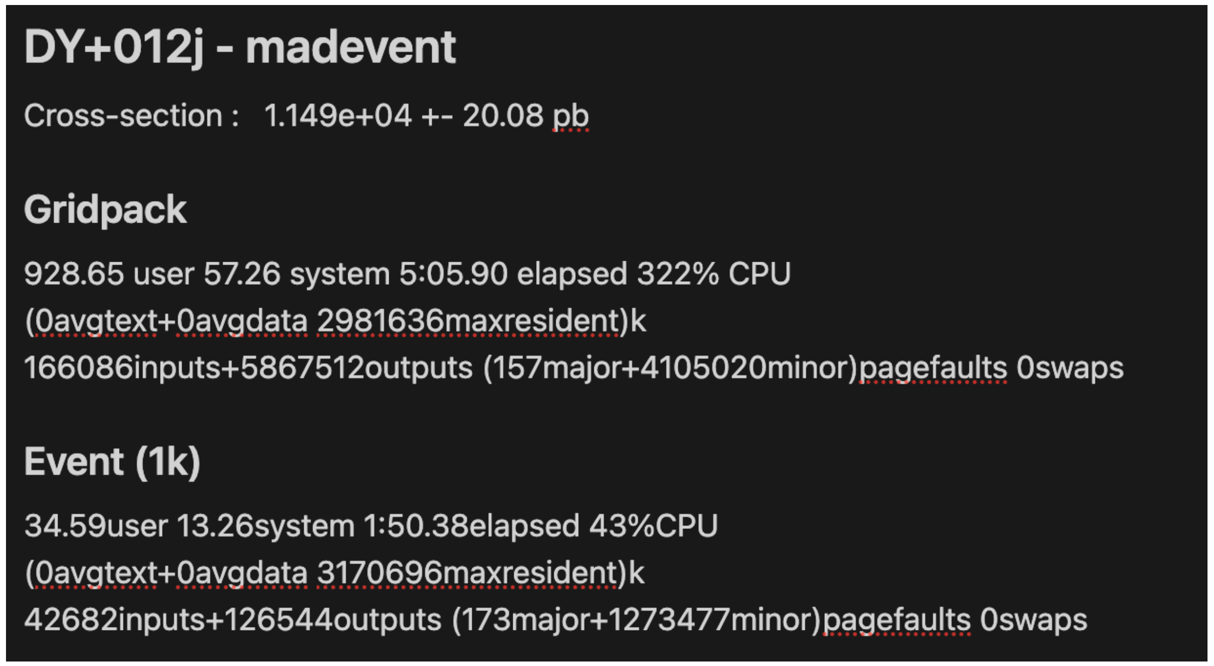


```
P0_dc_epemgdc
P0_dc_taptamgdc
P0_dd_epemgdd
P0_dd_taptamgdd
P0_ddx_epemgddx
P0_ddx_epemggg
P0_ddx_epemgssx
P0_ddx_epemguux
P0_ddx_taptamgddx
P0_ddx_taptamggg
```

…

TESTS

- **madevent vs. madevent_simd w/ fortran backend**
- DY+012j: 14 vs. 76
- Working environment: NERSC 128 CPUs + 4 A100 GPUs, nb_core = 16 for gridpack



DY+012j - madevent

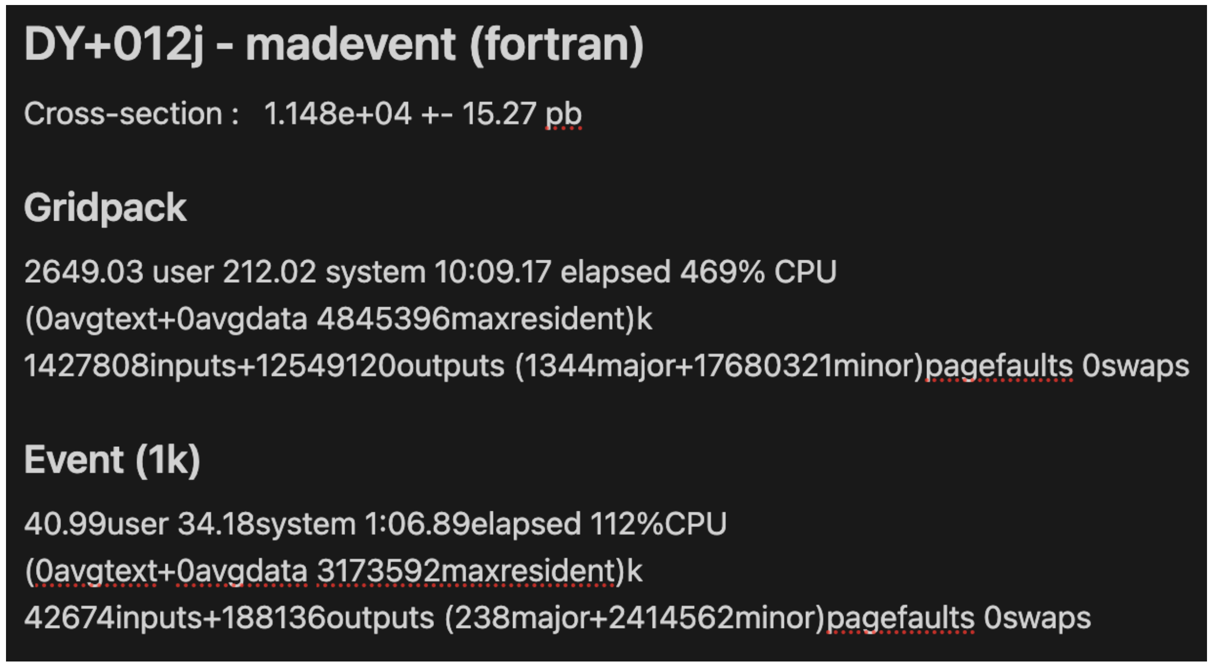
Cross-section : 1.149e+04 +- 20.08 pb

Gridpack

```
928.65 user 57.26 system 5:05.90 elapsed 322% CPU
(0avgtext+0avgdata 2981636maxresident)k
166086inputs+5867512outputs (157major+4105020minor)pagefaults 0swaps
```

Event (1k)

```
34.59user 13.26system 1:50.38elapsed 43%CPU
(0avgtext+0avgdata 3170696maxresident)k
42682inputs+126544outputs (173major+1273477minor)pagefaults 0swaps
```



DY+012j - madevent (fortran)

Cross-section : 1.148e+04 +- 15.27 pb

Gridpack

```
2649.03 user 212.02 system 10:09.17 elapsed 469% CPU
(0avgtext+0avgdata 4845396maxresident)k
1427808inputs+12549120outputs (1344major+17680321minor)pagefaults 0swaps
```

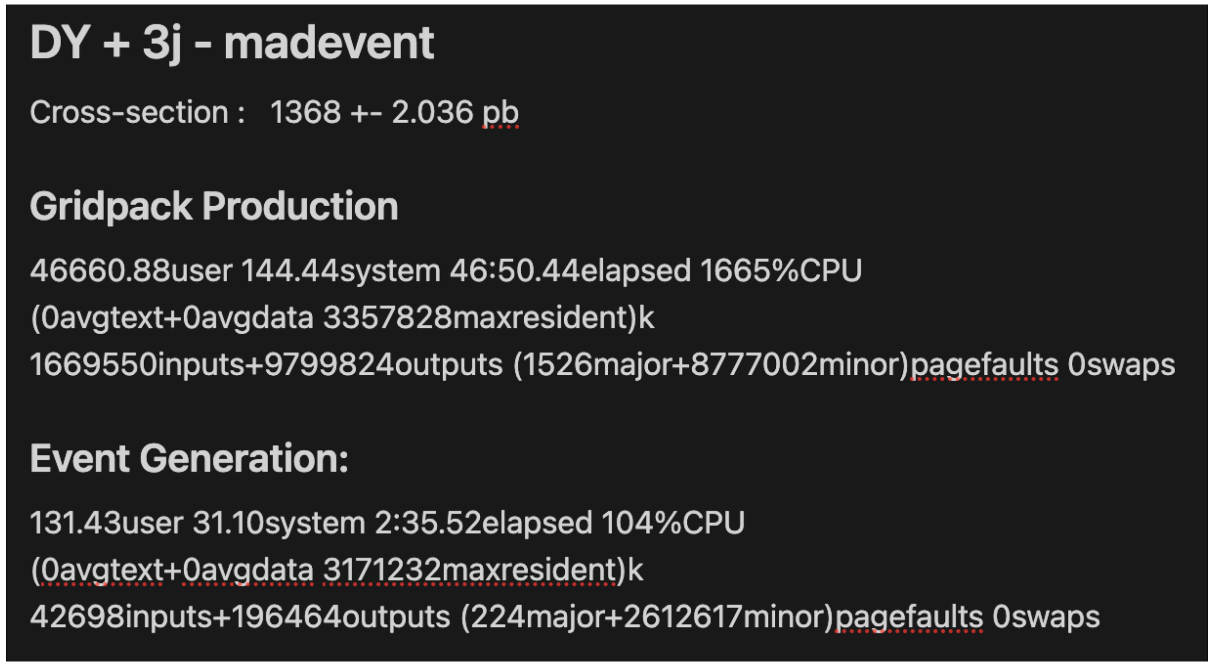
Event (1k)

```
40.99user 34.18system 1:06.89elapsed 112%CPU
(0avgtext+0avgdata 3173592maxresident)k
42674inputs+188136outputs (238major+2414562minor)pagefaults 0swaps
```

TESTS

madevent vs. madevent_simd w/ fortran backend

- DY+3j: 10 vs. 84
- Working environment: NERSC 128 CPUs + 4 A100 GPUs, nb_core = 40 for gridpack



DY + 3j - madevent

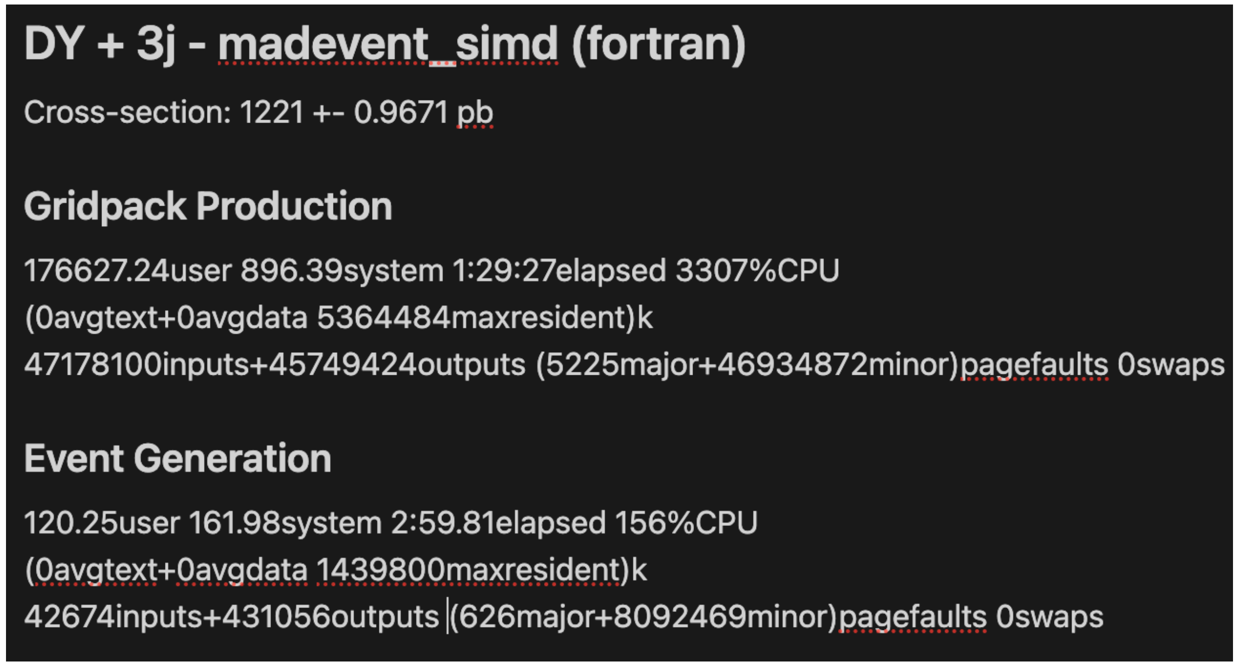
Cross-section : 1368 +- 2.036 pb

Gridpack Production

```
46660.88user 144.44system 46:50.44elapsed 1665%CPU
(0avgtext+0avgdata 3357828maxresident)k
1669550inputs+9799824outputs (1526major+8777002minor)pagefaults 0swaps
```

Event Generation:

```
131.43user 31.10system 2:35.52elapsed 104%CPU
(0avgtext+0avgdata 3171232maxresident)k
42698inputs+196464outputs (224major+2612617minor)pagefaults 0swaps
```



DY + 3j - madevent_simd (fortran)

Cross-section: 1221 +- 0.9671 pb

Gridpack Production

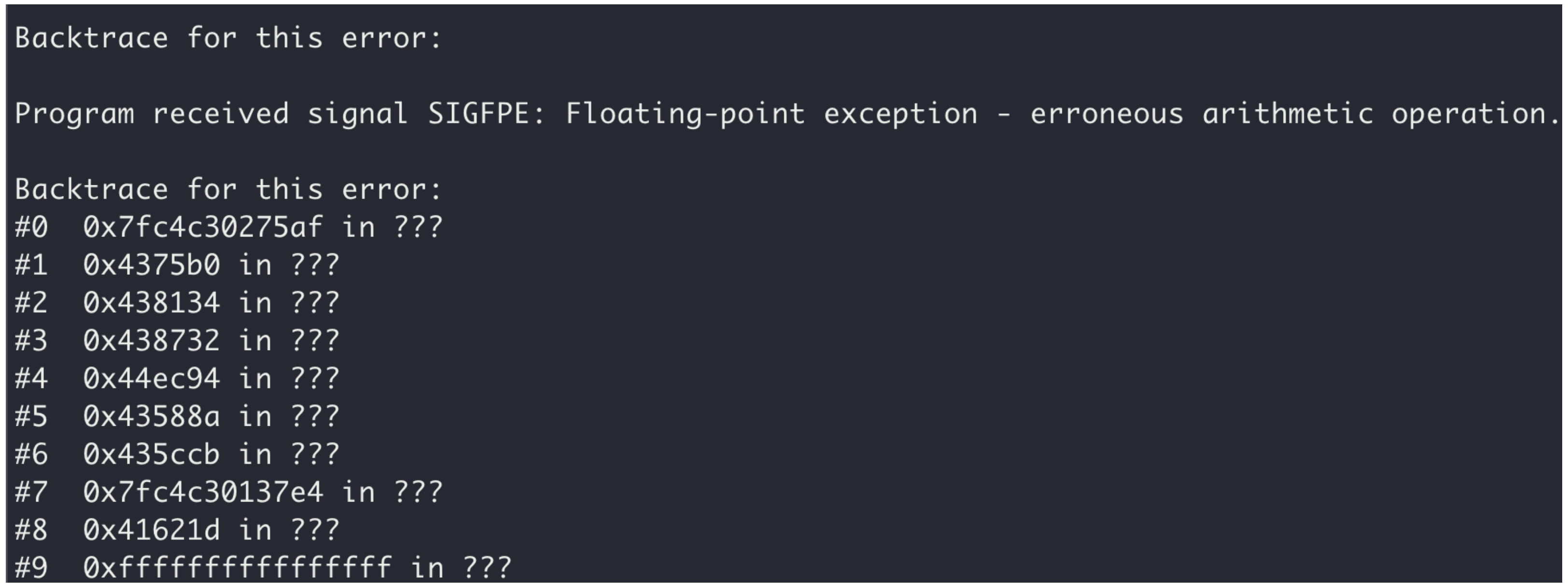
```
176627.24user 896.39system 1:29:27elapsed 3307%CPU
(0avgtext+0avgdata 5364484maxresident)k
47178100inputs+45749424outputs (5225major+46934872minor)pagefaults 0swaps
```

Event Generation

```
120.25user 161.98system 2:59.81elapsed 156%CPU
(0avgtext+0avgdata 1439800maxresident)k
42674inputs+431056outputs (626major+8092469minor)pagefaults 0swaps
```

ISSUES

- **Madspin integration test**
- ✓ gridpack production was fine for **fortran backend**
- **CPPAVX2 / CUDA test**
- ✓ Still facing FPE exceptions in CPPAVX2 and CUDA - need to be fixed before we moved on

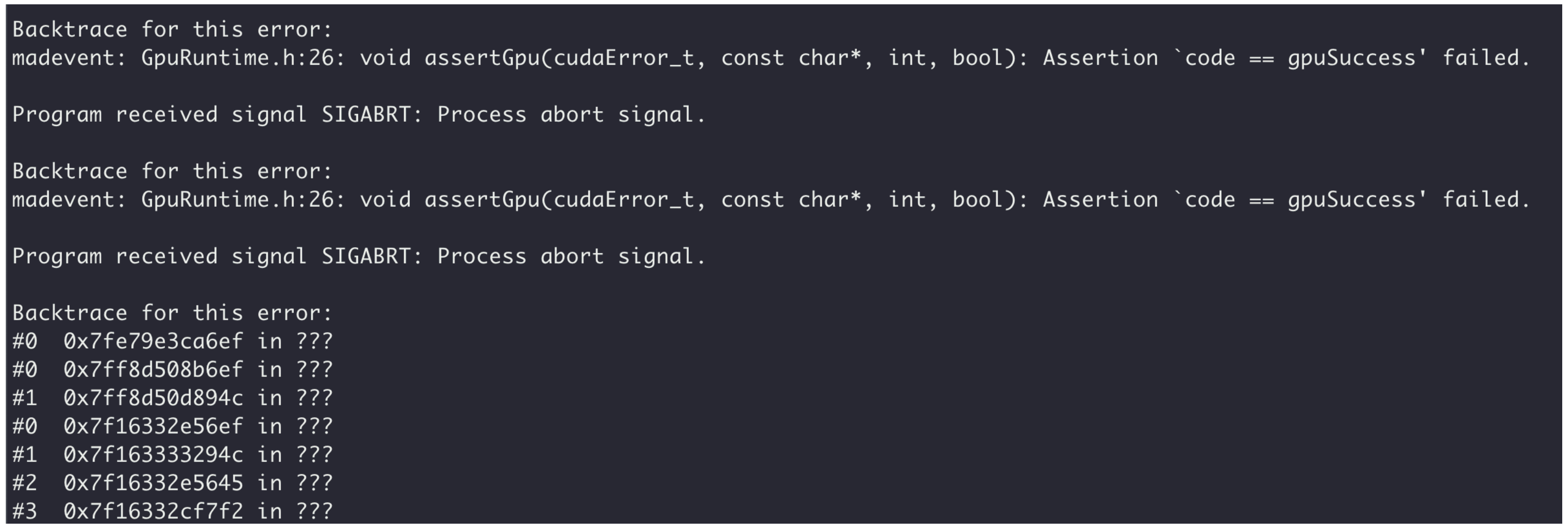


```
Backtrace for this error:

Program received signal SIGFPE: Floating-point exception - erroneous arithmetic operation.

Backtrace for this error:
#0  0x7fc4c30275af in ???
#1  0x4375b0 in ???
#2  0x438134 in ???
#3  0x438732 in ???
#4  0x44ec94 in ???
#5  0x43588a in ???
#6  0x435ccb in ???
#7  0x7fc4c30137e4 in ???
#8  0x41621d in ???
#9  0xffffffffffffffff in ???
```

- ✓ SIGABRT error in CUDA



```
Backtrace for this error:
madevent: GpuRuntime.h:26: void assertGpu(cudaError_t, const char*, int, bool): Assertion `code == gpuSuccess' failed.

Program received signal SIGABRT: Process abort signal.

Backtrace for this error:
madevent: GpuRuntime.h:26: void assertGpu(cudaError_t, const char*, int, bool): Assertion `code == gpuSuccess' failed.

Program received signal SIGABRT: Process abort signal.

Backtrace for this error:
#0  0x7fe79e3ca6ef in ???
#0  0x7ff8d508b6ef in ???
#1  0x7ff8d50d894c in ???
#0  0x7f16332e56ef in ???
#1  0x7f163333294c in ???
#2  0x7f16332e5645 in ???
#3  0x7f16332cf7f2 in ???
```