Progress report in the master branch
Initial analysis of the master_june24 branch

Andrea Valassi (CERN)

Madgraph on GPU development meeting, 08th July 2024
https://indico.cern.ch/event/1355156

(previous update was on June 25 – only mentioning changes since then)

Outline

- Progress in **madgraph4gpu master** (and **mg5amcnlo gpucpp**)
  - Our default branch, where I have shown all results in previous meetings
  - Fixes and improvements from myself and Olivier, with mutual reviews/help (thanks!)

- Progress in **madgraph4gpu master_june24**
  - A new parallel branch created by Stefan and Olivier for channelid #830 and warps #756
  - Associated to mg5amcnlo gpucpp_june24 and gpucpp_wrap branches (and lone commits)
  - Olivier asked me to look at this with high priority, towards a merge into master
    - (I used github fragments and reverse engineering – was this described in this meeting?)

master branch

Follow up on valgrind (see #868)

- Followed up the two issues I identified two weeks ago in testing madevent_fortran
  - (1) Minor leak in driver.f (file opened and not closed) mg5amcnlo#109
    - **Fix by AV (reviewed by OM) merged on Wed 26 Jun** in PR mg5amcnlo#110
  - (2) Uninitialized variable goodjet in reweight.f (possible undefined behavior) mg5amcnlo#111
    - **Workaround by AV (reviewed by OM) merged on Wed 26 Jun** in PR mg5amcnlo#112
    - Not a fix! Just 'randomly' initialize the uninitialized variables. A proper fix was also needed.
  - Update MG5AMC in madgraph4gpu accordingly (and regenerate code)
    - **Fix by AV (reviewed by OM) merged on Thu 27 Jun** in PR #869
  - After adding the patches for these two issues, valgrind is happy on madevent_fortran
    - HOWEVER: madevent_cpp was still crashing in rotxxx, no progress in this respect
  - (2') **Proper fix for goodjet by OM on Tue 2 Jul** in mg5amcnlo commit 1e2aa4bc3
    - Update MG5AMC in madgraph4gpu by AV on Wed 3 Jul as part of the color PR #877

- Tested madevent_cpp for 'segfault on Haswell' or 'out of bounds' reported by OM
  - I did not find any evidence of such issues (different input files? never mind...)

Major improvements in the CI

- Test coverage improvements
  - **Add my 'tmad' tests to the CI** (cross section and LHE file comparison Fortran-cudacpp)
  - Execute the full codegen-build-tput-tmad test chain for all FPTYPE=d,f,m
  - These tests were **ALREADY available to everyone** through my scripts, but a CI is better
    - Specific example/motivation: expose the rotxxx crash to everyone (no need for local reproducers)
- Test infrastructure improvements
  - Three separate jobs for FPTYE=d,f,m, reusing cached codegen from single codegen step
  - Separate build caches for the different FPTYPE's
  - Use the PR number in codegen and build cache lookup (separate caches of different PRs)
- **Patch by AV (reviewed by OM) on Thu 27 Jun** in PR #794

SIGFPE crash in rotxxx

- There is a SIGFPE crash #855 in Fortran function rotxxx (aloha_functions.f)
  - Only in optimized –O3 code: relevant variables in gdb show up as <optimized out>
  - Disabling optimization (IIRC –O1 is enough?) makes the crash disappear
  - My proposed workaround #857: add the *volatile* keyword for a few Fortran variables
    - Disable optimizations of very specific lines of code (related to Fortran SIMD?)
    - *This technique is extensively used in cudacpp SIMD ixx/oxx: volatile prevents many crashes*
  - Issue and fix are fully reproducible (crashes without, does not crash with volatile)
  - Not clear why it appears only for some iconfig – but I would fix this independently
    - And fixing this issue then makes it possible to see further issues down the line...

rotxxx crash #855 in gg_ttgg

```
Program received signal SIGFPE, Arithmetic exception.
rotxxx (p=..., q=..., prot=...) at aloha_functions.f:1247
1247                prot(1) = q(1)*q(3)/qq/qt*p1 -q(2)/qt*p(2) +q(1)/qq*p(3)
#0  rotxxx (p=..., q=..., prot=...) at aloha_functions.f:1247
#1  0x00000000004087e0 in gentcms (pa=..., pb=..., t=-181765.47706865534, phi=0.64468537567405615, ma2=0, m1=234.1712866912786,
    m2=210.15563843880372, p1=..., pr=..., jac=3.0327734872026782e+25) at genps.f:1480
#2  0x0000000000409849 in one_tree (itree=..., tstrategy=<optimized out>, iconfig=104, nbranch=4, p=..., m=..., s=..., x=...,
    jac=3.0327734872026782e+25, pswgt=1) at genps.f:1167
#3  0x000000000040bb84 in gen_mom (iconfig=104, mincfig=104, maxcfig=104, invar=10, wgt=0.03125, x=..., p1=...) at genps.f:68
#4  0x000000000040d1aa in x_to_f_arg (ndim=10, iconfig=104, mincfig=104, maxcfig=104, invar=10, wgt=0.03125, x=..., p=...)
    at genps.f:60
#5  0x0000000000045c865 in sample_full (ndim=10, ncall=32, itmax=1, itmin=1, dsig=0x438b00 <dsig>, ninvar=10, nconfigs=1,
    vecsize_used=16384) at dsample.f:172
#6  0x0000000000043427a in driver () at driver.f:257
#7  0x000000000040371f in main (argc=<optimized out>, argv=<optimized out>) at driver.f:302
#8  0x00007ffff743feb0 in __libc_start_call_main () from /lib64/libc.so.6
#9  0x00007ffff743ff60 in __libc_start_main_impl () from /lib64/libc.so.6
#10 0x0000000000403845 in  _start ()
```

- **Fixes by AV (reviewed by OM) merged on Thu 27 Jun**
  - Add volatile keyword in aloha_functions.f in PR mg5amcnlo#113
  - MG5AMC update in PR #857

Haswell segfault and LHE color mismatch

- Two related problems (wrong channel2iconfig mapping in coloramps.h impacts both)
  - OM reported a Haswell out-of-bound access segfault [AV could not reproduce this]
  - Color mismatch #856 in gg_ttggg for iconfig=104
  - *(NB: zero cross section in susy #826 is NOT related, even if a PR branch is fix_826...)*

- Two successive patches applied (after long useful discussion between AV and OM)
  - **Fixes by OM (reviewed by AV) merged on Wed 3 Jul** in "fix_826" PR #852
  - **Further patch by AV (reviewed by OM) merged on Wed 3 Jul** in PR #877
    - Replacing two older PRs #853 and #873 by AV as suggested by OM
    - Eventually stripping away AV's icolamp patch mg5amcnlo#116 for #856 as suggested by OM
  - These two patches fix channel2iconfig mapping in coloramps.h, but NOT yet icolamp
    - Consequence: #856 color mismatch is not yet fixed after these two patches

- A better fix for icolamp was eventually developed by OM (thanks Olivier!)
  - Using the findings of AV's icolamp patch mg5amcnlo#116, but much more robust
  - **Fix by OM (reviewed by AV) merged on Thu 4 Jul** in PR #880
  - **Further patch by AV (reviewed by OM) on Thu 4 Jul** in PR #881
  - These patches finally fix the LHE color mismatch #856

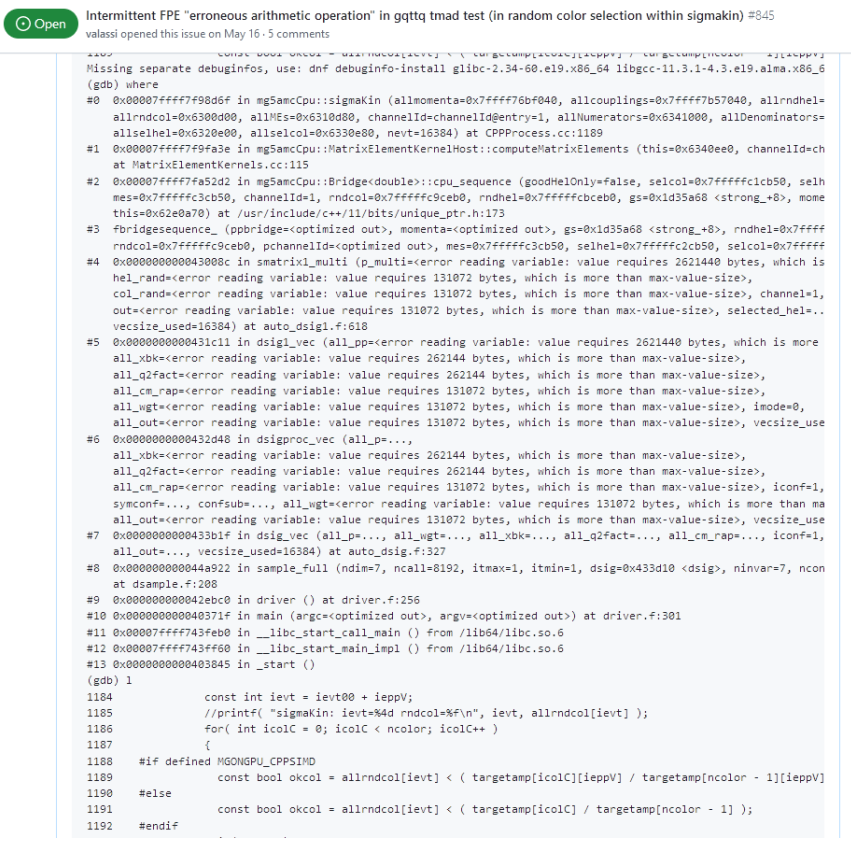
Disable gtest from launch

- Olivier's suggestion: disable googletest download/build/use from 'launch' #878
  - i.e. do not build cudacpp runTest.exe in 'user interface mode', also in generate_events
  - **Fix by AV (reviewed by OM) merged on Thu 4 Jul** in PR #879

AVX512 crash in gq_ttq color choice #845

- Clearly related to SIMD optimizations (only happens for 512z and FPTYPE=f)
  - Fixed again with a ‘volatile’ keyword...
  - **Fixes by AV (reviewed by OM) merged on Thu 4 Jul** in PR #874

- Fully reproducible with OpenMP disabled (was ‘intermittent’ with OpenMP enabled)
  - PR 874 also implements the option to disable OpenMP builds if ‘export OMPFLAGS=‘
  - Addresses suggestion by OM in #758
    - *TODO: switch off OMP by default?*


Open Intermittent FPE "erroneous arithmetic operation" in gqttq tmad test (in random color selection within sigmakin) #845
valassi opened this issue on May 16 · 5 comments

```
Missing separate debuginfos, use: dnf debuginfo-install glibc-2.34-60.el9.x86_64 libgcc-11.3.1-4.3.el9.alma.x86_6
(gdb) where
#0  0x00007ffff7f98d6f in mg5amcCpu::sigmaKin (allmomenta=0x7ffff76bf040, allcouplings=0x7ffff7b57040, allrndhel=
    allrndcol=0x6300d00, allMEs=0x6310d80, channelId=channelId@entry=1, allNumerators=0x6341000, allDenominators=
    allselhel=0x6320e00, allselcol=0x6330e80, nevt=16384) at CPPProcess.cc:1189
#1  0x00007ffff7f9fa3e in mg5amcCpu::MatrixElementKernelHost::computeMatrixElements (this=0x6340ee0, channelId=ch
    at MatrixElementKernels.cc:115
#2  0x00007ffff7fa52d2 in mg5amcCpu::Bridge<double>::cpu_sequence (goodHelOnly=false, selcol=0x7fffffc1cb50, selh
    mes=0x7fffffc3cb50, channelId=1, rndcol=0x7fffffc9ceb0, rndhel=0x7fffffcbceb0, gs=0x1d35a68 <strong_+8>, mome
    this=0x62e0a70) at /usr/include/c++/11/bits/unique_ptr.h:173
#3  fbridgesequence_ (ppbridge=<optimized out>, momenta=<optimized out>, gs=0x1d35a68 <strong_+8>, rndhel=0x7ffff
    rndcol=0x7fffffc9ceb0, pchannelId=<optimized out>, mes=0x7fffffc3cb50, selhel=0x7fffffc2cb50, selcol=0x7fffff
#4  0x0000000000043008c in smatrix1_multi (p_multi=<error reading variable: value requires 2621440 bytes, which is
    hel_rand=<error reading variable: value requires 131072 bytes, which is more than max-value-size>,
    col_rand=<error reading variable: value requires 131072 bytes, which is more than max-value-size>, channel=1,
    out=<error reading variable: value requires 131072 bytes, which is more than max-value-size>, selected_hel=..
    vecsize_used=16384) at auto_dsig1.f:618
#5  0x0000000000431c11 in dsig1_vec (all_pp=<error reading variable: value requires 2621440 bytes, which is more
    all_xbk=<error reading variable: value requires 262144 bytes, which is more than max-value-size>,
    all_q2fact=<error reading variable: value requires 262144 bytes, which is more than max-value-size>,
    all_cm_rap=<error reading variable: value requires 131072 bytes, which is more than max-value-size>,
    all_wgt=<error reading variable: value requires 131072 bytes, which is more than max-value-size>, imode=0,
    all_out=<error reading variable: value requires 131072 bytes, which is more than max-value-size>, vecsize_use
#6  0x0000000000432d48 in dsigproc_vec (all_p=...,
    all_xbk=<error reading variable: value requires 262144 bytes, which is more than max-value-size>,
    all_q2fact=<error reading variable: value requires 262144 bytes, which is more than max-value-size>,
    all_cm_rap=<error reading variable: value requires 131072 bytes, which is more than max-value-size>, iconf=1,
    symconf=..., confsub=..., all_wgt=<error reading variable: value requires 131072 bytes, which is more than ma
    all_out=<error reading variable: value requires 131072 bytes, which is more than max-value-size>, vecsize_use
#7  0x0000000000433b1f in dsig_vec (all_p=..., all_wgt=..., all_xbk=..., all_q2fact=..., all_cm_rap=..., iconf=1,
    all_out=..., vecsize_used=16384) at auto_dsig.f:327
#8  0x000000000044a922 in sample_full (ndim=7, ncall=8192, itmax=1, itmin=1, dsig=0x433d10 <dsig>, ninvar=7, ncon
    at dsample.f:208
#9  0x000000000042ebc0 in driver () at driver.f:256
#10 0x000000000040371f in main (argc=<optimized out>, argv=<optimized out>) at driver.f:301
#11 0x00007ffff743feb0 in __libc_start_call_main () from /lib64/libc.so.6
#12 0x00007ffff743ff60 in __libc_start_main_impl () from /lib64/libc.so.6
#13 0x0000000000403845 in _start ()
(gdb) l
1184            const int ievt = ievt00 + ieppV;
1185            //printf( "sigmaKin: ievt=%4d rndcol=%f\n", ievt, allrndcol[ievt] );
1186            for( int icolC = 0; icolC < ncolor; icolC++ )
1187            {
1188  #if defined MGONGPU_CPPSIMD
1189              const bool okcol = allrndcol[ievt] < ( targetamp[icolC][ieppV] / targetamp[ncolor - 1][ieppV]
1190  #else
1191              const bool okcol = allrndcol[ievt] < ( targetamp[icolC] / targetamp[ncolor - 1] );
1192  #endif
```

Pending physics issues in master

- No cross section in susy gg_t1t1 #826
  - OM/SR investigating if coupling ordering is responsible #862
- Cross section mismatch in pp_tt012j #872
- Much better than two weeks ago!!! Many crashes and complex issues fixed
- Both issues are now visible in the CI (6 failures: two times three FPTYPE's)
  - Note: the new CI has an option to bypass/ignore these two issues, check if enabled...

- *There is also WIP by OM on another CI extension, and some issues found there*
  - *Apologies, I did not have time to look at that yet...*

master_june24 branch

Merging master and master_june24

- Olivier asked me to look into the merge of master and master_june24
- I created my branch 'june24' in WIP PR #882
  - starting at master_june24
  - with the idea of progressively merging master into it
- To start with, I looked at master_june24 as-is, or with minimal modifications
  - I regenerated all processes with master_june24 codegen (so that the old CI can test them)
    - Processes had not been regenerated with the latest codegen, unlike what we had agreed long ago
  - I included the new CI tmad tests (so that any issues there immediately show up)
    - *NB: tmad tests (cross section and LHE comparisons) were ALREADY available via manual scripts*
- Many issues showed up, including *trivial build errors*, and crashes (see next slide)
  - **My opinion: channelid PR #830 was not sufficiently tested upfront**
    - Some issues may also arise because 'warp' modifications in mg5amcnlo for #765 are incomplete
  - We should agree on a procedure to avoid this happening again in the future...
    - (at the very least: regenerate the code and ensure all CI tests pass...)
- In the meantime, I am working on fixing these issues, so that we can move on...
  - I will come back to Stefan or Olivier when/if I have questions (I already asked some...)

Some of the issues in master_june24

- Builds fail for MAC/clang SIMD #883 (now fixed)
- Builds fail for FPTYPE=m #884 (now fixed)
- Crash in dsig1_vec if VECSIZE_USED < VECSIZE_MEMMAX #885
- Clarify gpucpp branches (master_june24 does not use gpucpp_june24) #886
- Fix documentation of NB_WARP and WARP_SIZE etc #887
- Fortran runtime error: index of array ‘symconf’ above upper bound #888
- Cross section mismatch for gg_ttggg FPTYPE=f #889 (now fixed, higher tolerance)
- Replace hack in counters.cc by a proper fix #891
- No-multichannel should be null array pointer, not channelid=0 #892
- __CUDACC__ macros prevent HIP support #893
- Wrong handling of SIMD numerators and denominators #894
- Channelid memory accessor should be called only once for all diagrams #895
- Add tests for two warps with different channelid’s #896

Is the warp functionality complete?

- Do I understand correctly that eventually one “Gn” job will handle many iconfigs?
- Is this functionality complete in Fortran? Is the new input.txt format decided?
- In the meantime: I would at least test two warps with different channelids #896

- *If warps are NOT complete, I would wait before merging master_june24 into master*
  - My opinion: we should make sure we are able to fully test the functionality

- In the meantime, the work on merging master into master_june24 can continue
  - And the work on master (cross sections, couplings, plugin...) should continue too