

Makefiles – two options forward?

- Option 1 (one-step) – start from Stephan’s PR [#753](#) (itself based on Jorgen’s [#775](#))
 - Advantages:
 - Stephan’s commits are immediately included
 - Disadvantages:
 - One-step approach mixes logically different features – debugging issues is more complex
 - Inherits issues to be fixed in Jorgen’s PR – disentangling and debugging issues is more complex
 - No HIP support until all makefiles issues are solved – will take longer, Nathan is waiting for it
- Option 2 (two-steps) – start from Jorgen’s [#775](#) first, later add Stephan’s PR [#753](#)
 - Advantages:
 - Debug issues in Jorgen’s PR separately from any issues in Stephan’s commits, logically cleaner
 - HIP support may be added before all issues in Stephan’s PR are fixed, Nathan may work earlier
 - Disadvantages:
 - Must fix conflicts in the merge, or (better) add back Stephan’s features one by one
- I would most certainly go for option 2 – technically more appropriate

About Stephan's PR #753 (1)

- I started analyzing and fixing issues in my PR #797 (I need a PR to test the CI...) – (I had understood Stephan asked me/us for help at some point, especially for SA?)
- The CI tests for madevent fail (manual tests failed too) – I started from here (Dec 17)
 - Manual codegen failed on el9 (clang-format issues) and el8 (patchMad.sh issues)
 - Fixed by adding clang-format-off and by handling f2py3 for make_opts in patchMad.sh
 - But the CI tests still fail after fixing codegen on manual el8/el9 tests
 - The CI fix initially broke manual el9 again, but eventually all fixed (reorder variables in make_opts)
 - NB: the PR handles make_opts in patchMad.sh, I would rather avoid that (and OM agrees?)
 - Next issue: 'make clean' fails if called twice
 - Fixed by changing the bash command so that it does not rely on nullglob behavior
 - Next issue: 'make -j' may fail (peculiar case but I was unlucky – CTRL-C then restart)
 - Fixed by improving the dependency on discretessampler.mod
 - Next item: analyze AVX targets... there are some things I would do differently
 - Here I realized these 'issues' were inherited from Jorgen's PR and I got the idea of looking at that
- The CI tests for SA fail – I only looked at this later on (Dec 18)
 - Here I realized that some lines (AVX and more) from cudacpp.mk have been moved to make_opts, which is not (yet) included in cudacpp.mk... so SA is bound to fail
 - And again, the AVX stuff I would clean up differently in Jorgen's PR first, independently

About Stephan's PR #753 (2)

- A few random suggestions on how to do this better maybe?
 - If some common stuff is needed in cudacpp.mk and cudacpp_src.mk, put it in a new cudacpp_common.mk, not in make_opts
 - Avoid cluttering make_opts with stuff that is fully internal to cudacpp
 - Even better: I had already started cleaning up the duplicates between cudacpp.mk and cudacpp_src.mk by using 'export' of variables
 - Do not remove all overrides: some of them indicate variables that should not be modified from the outside (would be 'private' stuff in C++...)
 - I would still call 'make -f cudacpp.mk <target>' inside the madevent makefile, rather than including cudacpp.mk inside makefile
- Rephrasing, I would
 - Look at Jorgen's PR first and fix all AVX issues
 - Remove duplicates between cudacpp.mk and cudacpp_src.mk using export
 - Keep 'make -f cudacpp.mk, clean up instead my ugly build dir variable retrieval
 - Only then, start looking at which variables we want to set from outside or from make_opts
 - Only then, actually include make_opts and see what it gives

About Jorgen's PR #775

- After stopping work over Stephan's PR #753, I looked at Jorgen's #775
- This also has some issues to be fixed (or in my opinion improved)
 - [had no time to re-analyse all this in detail]
 - I rebased on upstream/master
 - I replaced the use of different targets (which I had proposed to Jorgen, mea culpa) by the use of `BACKEND=cuda/512y/...` (similar to what was previously done with AVX)
 - Much more similar to current upstream/master
 - Removes the need to rely on `MAKECMDGOALS`
 - I also fixed 'make cppall' which had issues
 - I exported `AVXFLAGS` (as done with other variables) to avoid duplication `cuda/cudacpp/cudacpp_src`
 - Improved handling of `MadgraphTest.h` and fixed `runTest.exe` build
 - Also using some commits or code snippets from Stephan
 - Adapted and rerun all `tput/tmad/tlau` tests
- Essentially complete, except for
 - Some integration with `runcards` to be completed (I was waiting for go ahead)

Recent open PRs (beyond makefiles)

- From scalar channel ID to array of channel IDs (issue [#765](#))
 - Stefan's [new interface wrap](#) branch (no PR yet): status?
 - To add: Andrea's [fbridgesequence_multichannel](#) in PR [#796](#) (review requested Nov 23)
 - Eventually need also Olivier's [mg5amcnlo gpucpp wrap](#) (not yet in gpucpp): complete?
- HIP support for AMD GPUs (issue [#311](#))
 - Part 1: separate CUDA and C++ targets in Makefiles (issues [#602](#), [#680](#))
 - Jorgen's PR [#775](#), with Andrea's merge of master and other fixes in PR [#798](#)
 - Part 2: add GpuAbstraction.h
 - Jorgen's PR [#774](#), but still need merge of master
 - Part 3 to do: add HIP make targets, set up AMD GPU nodes for CI and interactive tests
 - Jorgen did tests on LUMI in PR [#718](#), but we need to check our nodes again
 - Put it all together and test the whole thing
- Cleanup: remove gXXX.cu symlinks
 - Andrea's PR [#378](#) (review requested Nov 22, waiting for other makefile changes)

Other missing items before a release

- Update the plugin (issue [#661](#)) – or recreate it with the full history?
 - [mg5amcnlo_cudacpp](#) exists with the full history but is stuck to Aug 30
- Do we want to try and fix some EFT and SUSY before the release?
- See also the May 2023 summary (issue [#671](#)): not up to date but still relevant
 - The issues that are still open remain desirable, though not strictly necessary
 - Many issues mentioned there have been fixed/completed
 - Only a few new issues have appeared (e.g. channel id array)
- Am I missing some very big thing not mentioned above?

Backup (older version of slides)

Recent open PRs (makefiles)

- For reference, previous slide:

- HIP support for AMD GPUs (issue [#311](#))
 - Part 1: separate CUDA and C++ targets in Makefiles (issues [#602](#), [#680](#))
 - Jorgen's PR [#775](#), with Andrea's merge of master and other fixes in PR [#798](#)

- There are various feature needs/wishes
 - Separate C++ and CUDA (and later HIP!) targets and builds – Jorgen's work
 - General refactoring of makefiles – Stephan's work
 - “Allow setting flags from outside” (AV: OK, at least for some flags)
 - “Move common setup to make_opts” (AV: do we want this? check with OM)
 - “Include rules from C++ makefile into Fortran” (AV: no, keep cudacpp as a library? – also for SA)
 - A few other improvements, e.g. define a “native” arch build for SIMD (AV: OK)
- There is a chain of PRs and branches, but I would keep the features separate!
 - Jorgen's original PR [#775](#) is only for separating CUDA and C++
 - Andrea's PR [#798](#) (~almost complete: tests pass) over Jorgen's original PR [#775](#)
 - keep only this feature; in addition merge latest master and add a few fixes/improvements (to be discussed)
 - Stephan's PR [#753](#) over Jorgen's original PR [#775](#)
 - add general refactoring of makefiles: has some issues as Jorgen's plus a few extra own issues (tests fail)
 - Andrea's PR [#797](#) over Stephan's PR [#753](#)
 - I tried to fix the CI tests, then I tried to fix SA... but I realised that this is just by design impossible now
 - I suggest doing HIP stuff first, and any further refactoring only in a later step (while keeping SA working)